INTERACTIVE SYSTEMS RESEARCH

M. I. Bernstein

System Development Corporation

Prepared for:

Advanced Research Projects Agency

15 May 1975

INTERACTIVE SYSTEMS RESEARCH: INTERIM REPORT TO THE DIRECTOR, ADVANCED RESEARCH PROJECTS AGENCY, FOR THE PERIOD 16 SEPTEMBER 1974 to 15 MARCH 1975

15 MAY 1975

THIS REPORT WAS PRODUCED BY SDC IN PERFORMANCE OF CONTRACT DAHC 15-73-C-0080, ARPA ORDER NO. 2254, PROGRAM CODE NUMBER 5D30.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. 3. RECIPIENT'S CATALOG NUMBER AD-A012 [illegible]
4. TITLE (and Subtitle) Interactive Systems Research: Interim Report to the Director, Advanced Research Projects Agency, for the Period 16 September 1974 to 15 March 1975	5. TYPE OF REPORT & PERIOD COVERED Technical 16 Sept 74 - 15 March 75
	6. PERFORMING ORG. REPORT NUMBER TM-5243/003/00
7. AUTHOR(s) Bernstein, M. I.	8. CONTRACT OR GRANT NUMBER(s) DAHC15-73-C-0080
9. PERFORMING ORGANIZATION NAME AND ADDRESS SYSTEM DEVELOPMENT CORPORATION 2500 Colorado Avenue Santa Monica, California, 90406	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5D30
11. CONTROLLING OFFICE NAME AND ADDRESS Information Processing Techniques Office (IPTO) Advanced Research Projects Agency Arlington, Virginia	12. REPORT DATE 15 May 1975
	13. NUMBER OF PAGES 75
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Cleared for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

PRICES SUBJECT TO CHANGE

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acoustic-phonetics	Data management	Speech understanding
ARPA Network	Lexical semantics	
Artificial intelligence	Linguistic processing	
Computer networks	Prosodics	
Data-base conversion	Semantic analysis	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report summarizes six months of research activity in the development of three computer hardware/software systems. The largest one is being designed to respond to data management queries given to it by users via natural English spoken into a microphone and digitized for acoustic-phonetic and linguistic processing; completion and demonstration is scheduled for December 1976. A second and related system is one in which lexical semantic information on English words is being archived as a resource for the ARPA speech understanding research [illegible]nity and others. The third is a general-purpose data-base-

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (continued)

conversion system designed for use with existing data management systems without requiring any modifications to those systems. The ARPA Network is being used as a resource in these three activities.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION AND SUMMARY

This report to the Advanced Research Projects Agency (ARPA) is a summary of research progress during the first six months of System Development Corporation's current twelve-month contract for a program of Interactive Systems Research. The program presently includes three projects: (1) Speech Understanding Research, (2) Lexical Data Archive, and (3) Common Information Structures. The overall intent of the research is to provide bases for improved man-machine interactive systems and for new data management system capabilities. The major emphasis during the past two years has been on speech understanding, on facilities for supporting the speech understanding research community, and on data base conversion.

Our work in speech understanding research contains many developments that will be of material aid in enhancing and improving interactive systems by enabling them to be more responsive to the casual user. The continuing effort to permit such users to easily and effectively communicate with computer-based systems in language forms that are natural to them is of particular importance. An important element in making this possible is the considerable advancement in our understanding of the basic processes involved in natural language understanding that the present research is providing.

In support of ARPA's Speech Understanding Research program and other language-based efforts, the Lexical Data Archive project was started to create (as its name implies) a centrally organized archive of lexical semantic information for, and of particular use to, all ARPA contractors working in speech understanding, as well as other language researchers.

The Common Information Structures project is developing techniques for semi-automatic conversion of large data bases from one hardware-software environment into another with minimum effort, cost, and disruption to users. With the continuously evolving environments of new hardware, new operating systems, and new data management systems, such techniques are becoming as important as smooth, well-engineered user interfaces, and we see our efforts in both of these areas as complementary.

The following paragraphs summarize these projects' activities during the past six months.

Speech Understanding Research

During the previous contract year, SDC joined forces with Stanford Research Institute (SRI) to produce a mid-range prototype of the so-called "five-year" speech understanding system first described by an ARPA Study Group in 1971. This prototype, and further versions of it that will lead to the demonstration of the five-year system in 1976, capitalizes on the strength of SDC's advanced technology in acoustic-phonetic signal processing and large-scale system integration and on SRI's advanced work in higher-level linguistic processing,

including syntax, semantics, discourse analysis, and the integration of these elements into parsing strategies specifically designed for speech-input analysis. Extensive testing of the prototype, along with several experiments and a large number of protocol studies, indicated significant improvements to the system that would lead to an expanded and more powerful system targeted for implementation in September, 1975. The capabilities of that system, which we call the Milestone System, are described in some detail in this report. Briefly, it will contain a vocabulary of 600 words (leading to the vocabulary of 1,000 words for the five-year system) having to do with data on the naval fleets of the United States, the United Kingdom, and the USSR. It will operate on a configuration of three computer systems: a PDP-11/40 and an SPS-41, which will do the front-end acoustic-phonetic processing of the speech input, and an IBM 370/145, which will perform the higher-level linguistic processes being developed by SRI. It will respond to both male and female speakers, and response will be in approximately 25 times real time. The system will be extensively tested during the early part of the subsequent year in order to develop the refinements and enhancements that will be necessary for the five-year system.

Lexical Data Archive

Files with a considerable amount of semantic lexical data on words in the ARPA SUR system lexicons are now available over the ARPA Network. During the past six months, implementation of additional files was begun, a system of user manuals was developed, and more powerful formalizations of many of the files were implemented. Significant amounts of effort were devoted to preparing semantic analyses of words in the SUR lexicons and to developing appropriate file-management and access facilities for the data in the archive.

Common Information Structures

In its third year, the Common Information Structures project has begun to implement a system of languages and translation modules that will permit the transfer of data bases between any two differently structured data management systems, even those that reside on different computer systems. The major advantage of this system over previous designs is that the query and generate functions of the two data management systems are incorporated into the conversion process, permitting it to be much simpler and more economical. Having developed the language design during the previous contract year, the project has concentrated during the first half of the present year on the specification and implementation of the translation modules and on formalizing the types of conversion mappings that will be required for a variety of existing data management system types.

1. SPEECH UNDERSTANDING RESEARCH

1.1 INTRODUCTION

Late in 1974, the first fully integrated prototype of a speech understanding system developed jointly by SDC and the Stanford Research Institute (SRI) was implemented. The task domain of the system is data management on attributes of submarines. The system operates on SDC's Raytheon 704 and IBM 370/145 computers. The Raytheon computer is used to perform an acoustic-phonetic analysis of a digitized speech waveform. The results of this analysis are put into an array of acoustic-phonetic data that we refer to as the A-matrix. The data in the A-matrix are used by lexical mapping procedures to verify the existence of words hypothesized by a "best-first" parser that draws on a set of language-definition (syntax) rules and on components containing semantic and pragmatic (discourse-context) sources of knowledge.

The acoustic-phonetic processing and lexical mapping procedures are essentially the same as those used by SDC in its Voiced-controlled Data Management System (see, e.g., Ritea [19,20]), modified to handle a vocabulary of 300 words. The system now has a word-string mapping procedure that handles coarticulation between pairs of words.

The parser is a major revision and extension of the previous SRI parser (Walker [22], Paxton [18]), in which sources of knowledge are separated from the procedures for applying them. A best-first strategy still prevails, but it is now possible to start from any fixed point in the utterance, to skip over portions, and to accept input from word-spotting routines. The syntax encompasses that of the previous SRI system but has been extended to cover isolated noun phrases and nominals. In addition to being independent of the parser, it has been rewritten as a series of context-free rules with factors that specify restrictions or conditions on rule application; as a result, it can be used top-down, bottom-up, or with missing segments. The semantic component has been completely revised. Information is now stored in a network representation; corresponding to each syntactic rule is a semantic interpretation rule that operates on the network. A pragmatic component, based on analysis of protocol studies, has been added to handle anaphora and ellipsis and to provide discourse constraints for processing dialogue dependencies.

Ten utterances were selected for initial system testing and checkout. They are shown in Table 1-1. Results from processing these utterances are now being used to guide system debugging. Testing and debugging will continue until a stable, reliable system is obtained.

The goal for the end of this contract year (September, 1975) is the Milestone System. The task domain for the Milestone System will be data management with an expanded data base containing attributes of submarines, aircraft carriers,

Table 1-1. Test Utterances

1. What is the surface displacement of the LaFayette?
2. The Ethan Allen?
3. Submerged Displacement?
4. What is the speed of it?
5. How many Guppy 3s do we have?
6. How many Lafayettes?
7. The U.S. has Lafayettes.
8. Who has Nucs?
9. What submarines have a length of more than 400 feet?
10. The Sea Wolf has six torpedo-tubes.

and ocean escorts of the U.S., U.S.S.R., and U.K. The vocabulary will be extended to 600 words; the system will accommodate six speakers, both male and female. Response to the user will be about 25 times real-time. The operating environment will have a signal/noise ratio of about 30-40 dB. Acoustic analysis will be performed on PDP-11/40 and SPS-41 computers, which will be interfaced to the IBM 370/145. Refined and augmented acoustic-phonetic analysis will include improved formant tracking, pitch tracking, and vowel-sonorant analysis. Techniques will be developed for handling voiced fricatives and plosives, and improvements will be made to the present programs for handling unvoiced fricatives and plosives. A new programming system, CRISP, will provide efficient arithmetic and array processing, in addition to efficient symbol and list processing, and will substantially increase the address space. Two new mapping procedures will have been added: one to handle prosodic features and one to provide word spotting and do lexical subsetting on the basis of robust acoustic cues. The only major change expected in the parser is increased efficiency, which will be made possible through conversion to CRISP. Modifications to the syntax will include the addition of time and place, the conjunction and negation of noun phrases, the use of prosodic attributes and factors (already being written into the rules), and the capability of incomplete sentences to function as utterances. Semantic information will guide retrieval and prediction in addition to interpretation. The discourse model will be augmented to handle longer dialogue sequences on the basis of protocols gathered in more carefully controlled experiments. System exercising will be guided by formal test and validation procedures that will assess each component's contribution to system performance.

1.2 PROGRESS AND PRESENT STATUS

The following sections outline the progress made by SDC during the past six months in the development of the acoustic-phonetic processor, lexical mapping procedures, and the system hardware and software that will be used for the

Milestone System. In addition, a description is given of the development of our protocol experiments and prosodic analysis, which are tasks we share with SRI.

1.2.1 Acoustic-Phonetics

To date, we have made progress in the following areas of acoustic-phonetic analysis:

(1) fundamental frequency extraction;

(2) RMS energy and silence detection;

(3) formant frequency analysis;

(4) vowel and sonorant analysis;

(5) lateralization; and

(6) isolation and characterization of fricatives and plosives.

These are discussed in turn below.

Fundamental Frequency Extraction

The speech signal is first digitized at the rate of 20,000 samples per second. The fundamental frequency extraction program [6] operates on the digitized speech in three phrases:

(1) Down-sampling. A digital filter is used to reduce the sample rate of the speech from 20,000 samples per second to 2,000 samples per second, thus removing many frequencies that lie outside the range of possible fundamentals and improving the speed of the program.

(2) Autocorrelation and pitch extraction. An autocorrelation spectrum with a window size of 50 ms. is taken every 10 ms. An algorithm examines these spectra and picks peaks from them; it also edits out octave errors (mistaking a harmonic or subharmonic for the fundamental). To eliminate frequency quantization, a parabola is fitted to the peak chosen in the spectrum, and the theoretical peak of this parabola is used as the pitch value.

(3) Editing. The values of fundamental frequency obtained above are passed through a median smoother to eliminate anomalous values, and then a heuristic pitch-track editor attempts to remove any remaining errors. The pitch tracker enters a fundamental frequency estimate for each frame having a pitch.* These frames are labeled as voiced. All other frames are labeled unvoiced.

*A frame is a 10-ms. interval of the digitized speech waveform. It is the basic unit from which the A-matrix is constructed.

RMS and Silence Detection

An RMS engergy parameter is calculated for each frame; this parameter is a measure of the total energy from 0 to 10,000 Hz for that frame. The unvoiced frames are then subjected to a test to determine whether they can be labeled as silence.

A digital filter technique is used to produce a zero-crossing count for each frame. The zero-crossing information is later used to differentiate voiced strident fricatives from other voiced sounds.

Formant Frequency Analysis

LPC spectra are taken for each voiced frame, and for unvoiced frames that are within five frames of a voiced frame (providing there are no intermediate silence frames). The LPC spectrum is taken with a 25.6 ms. Hamming window centered at the middle of the frame. (The LPC technique used is taken from Markel et al. [10].) A peak-picker is used to extract the formant information. A maximum of five robust frequency peaks, arbitrarily labeled FA through FE, are selected, along with amplitudes and bandwidths. The frequency of the highest peak must be below 5,000 Hz. Inflection points in the spectrum may be selected as peaks, and, when this happens, each occurrence is flagged for later use by the formant tracker. The peak-picker is not a passive process; it makes heuristic judgments. No attempts at peak or formant tracking are made in the peak-picker, nor is any attempt made to place peaks within certain frequency ranges.

The peak picker begins by building first-and-second-difference frequency tables. By inspection of these tables, all peaks and inflection points are located in the 0-5 kHz spectrum. If an isolated large-bandwidth peak is found, an off-axis spectrum is calculated in an attempt to resolve the peak into two peaks. (Figure 1-1 illustrates an example of two formants appearing as one peak in the LPC from the "g" in ago.) If the total number of peaks and inflection points is greater than five, an off-axis spectrum is also calculated in an attempt to remove extraneous inflection points.

Next, the formant tracker selects the most likely F1, F2, F3, and F4 frequencies from the FA through FE formant information provided by the peak picker. Figure 1-2 illustrates the four steps in the formant tracking of the words "we hear the boy" taken from a longer utterance. Step 1 is a dot-plot of the formant frequency output of the peak picker. In Step 2, the formant tracker begins by moving from left to right and linking frequencies of adjacent segments that are within a threshold difference of each other (the link is not made if a frequency in one segment could be linked to more than one frequency in the adjacent segment). The solid lines in the Step 2 illustration indicate where the links have been made.

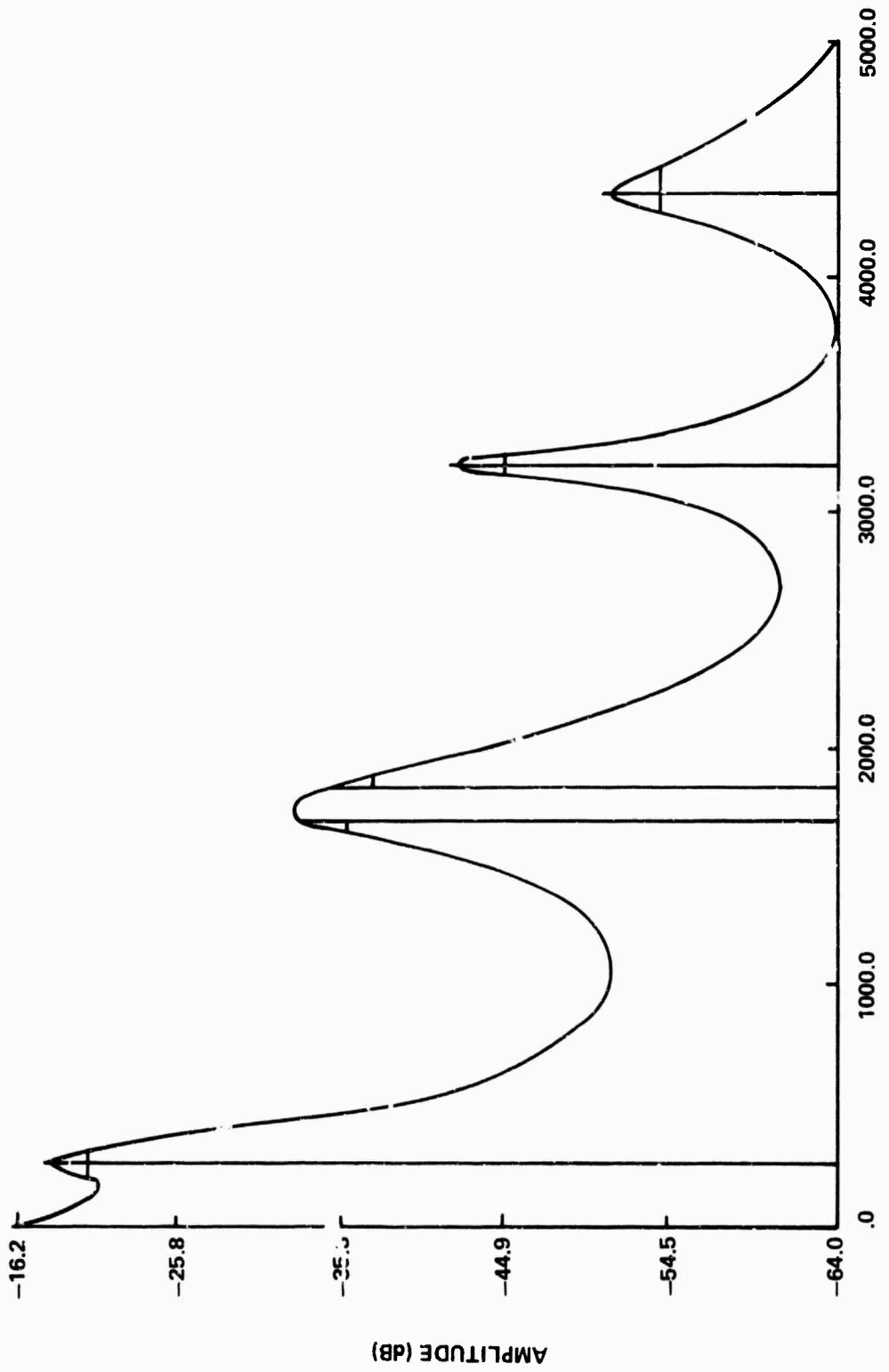


Figure 1-i. Two Formants Appearing as One Peak

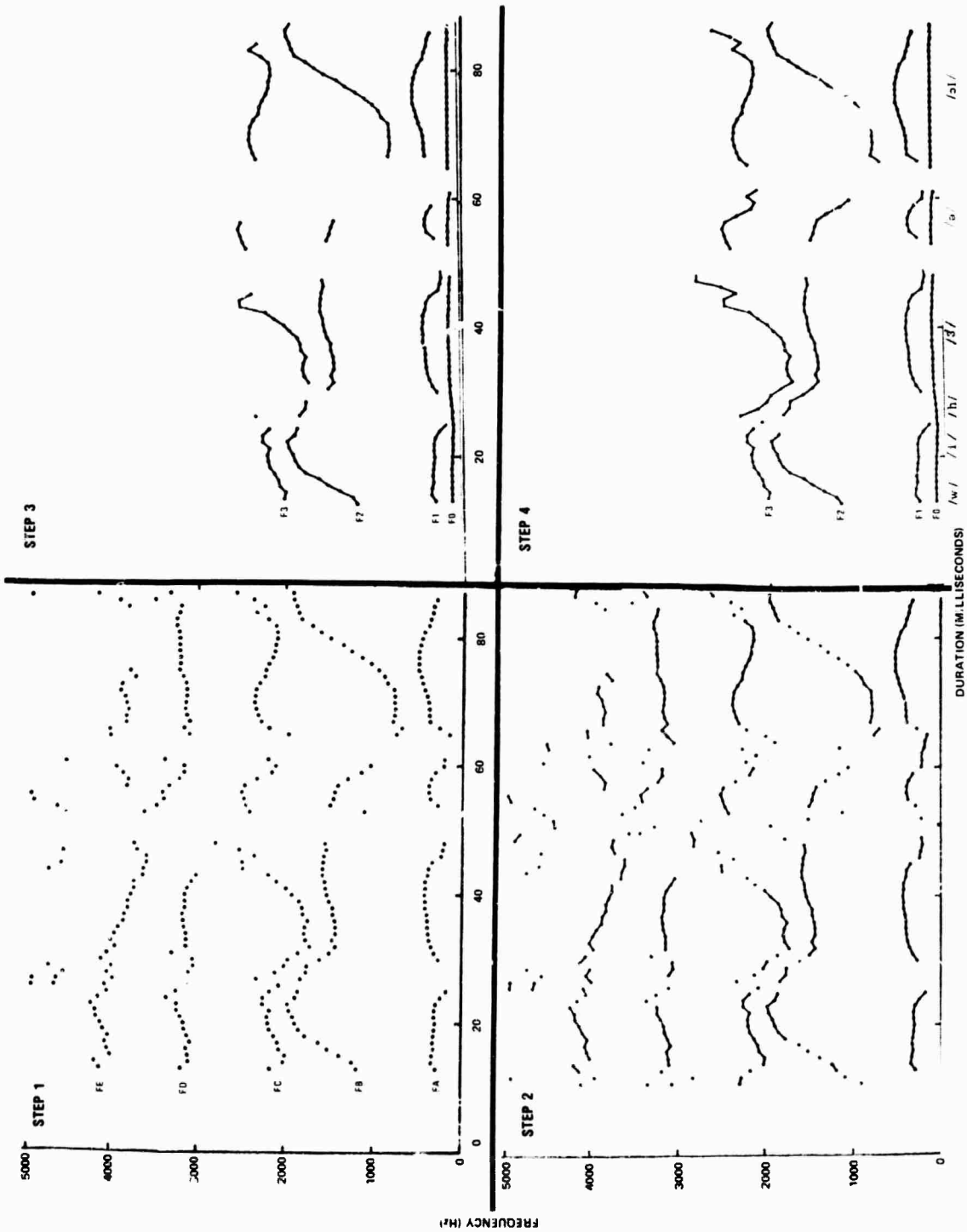


Figure 1-2. Example of Formant Tracking Steps

In Step 3, anchor points for finding F1 through F4 are selected on the basis of the FA through FE links. (F4 is tracked because it aids in defining F3 in areas where F3 is moving rapidly.) Each two adjacent segments in which each segment has at least four links are selected as anchor points. For each anchor point, this yields three consecutive linked segments for linking F1 through F4. The decisions make use of speaker-dependent vowel-sonorant tables in fixing minima and maxima for the formant frequencies. (The minima and maxima are used by the program as reference points; assigned formant-frequency values may be outside the limits if certain criteria are met.) Where choices need to be made between frequencies, relevant information such as bandwidths, amplitudes, indicators denoting points of inflection, and local RMS dip information is utilized. If an assignment cannot be made easily, the formant is left unassigned.

After all the anchor points have been fixed, an iterative process for assigning F1 through F4 parameters is begun. The labels F1 through F4 are extended to the frequencies linked to the right and left of each anchor point, and then missing formants are filled in where they are obvious. This is done iteratively until no further assignments can be made; the results are shown in Step 3. The bottommost track shown in Step 3 is that of the fundamental frequency (FO).

In Step 4, segments that have unassigned formants are filled in. This is accomplished primarily by moving from voiced areas with assigned formants into the adjacent areas with unassigned formants. In addition: if (1) a formant peak has disappeared for one or two consecutive segments in a voiced area (as shown in Figure 1-3 at time 120 centiseconds and again at 169 and 172 centiseconds), and (2) the difference in formant frequency between the right boundary and the left is within a threshold, and (3) there is only one formant value missing in the segment, and (4) there is no ambiguity in filling in the missing formant, then a linear interpolation is made and the formant frequency value calculated by the formant tracker is entered, and an indication of such is made in the A-matrix.

When these processes are completed, the formant tracker enters the frequency, amplitude, and bandwidth information for F1 through F3 into the A-matrix. It also stores the same information for F4 or a nasal formant if either are available or appropriate.

Discontinuities that now appear due to transition to or from unvoiced areas will be reexamined and corrected in preparing for the Milestone System. Figure 1-4 illustrates a discontinuity that occurred after the /p/ in "pretty" because F3 disappeared for a segment and the real F4 had been assigned to the F3 slot. Figure 1-5 shows discontinuities between a nasal and a vowel or sonorant area. This is seen on the graph as the /n/ in "lion roar" near centisecond 170. These discontinuities can be detected by the computer if it realizes that there is continuity in the voiced areas on both sides of the discontinuity and that one of the voiced areas fits a nasal pattern such as

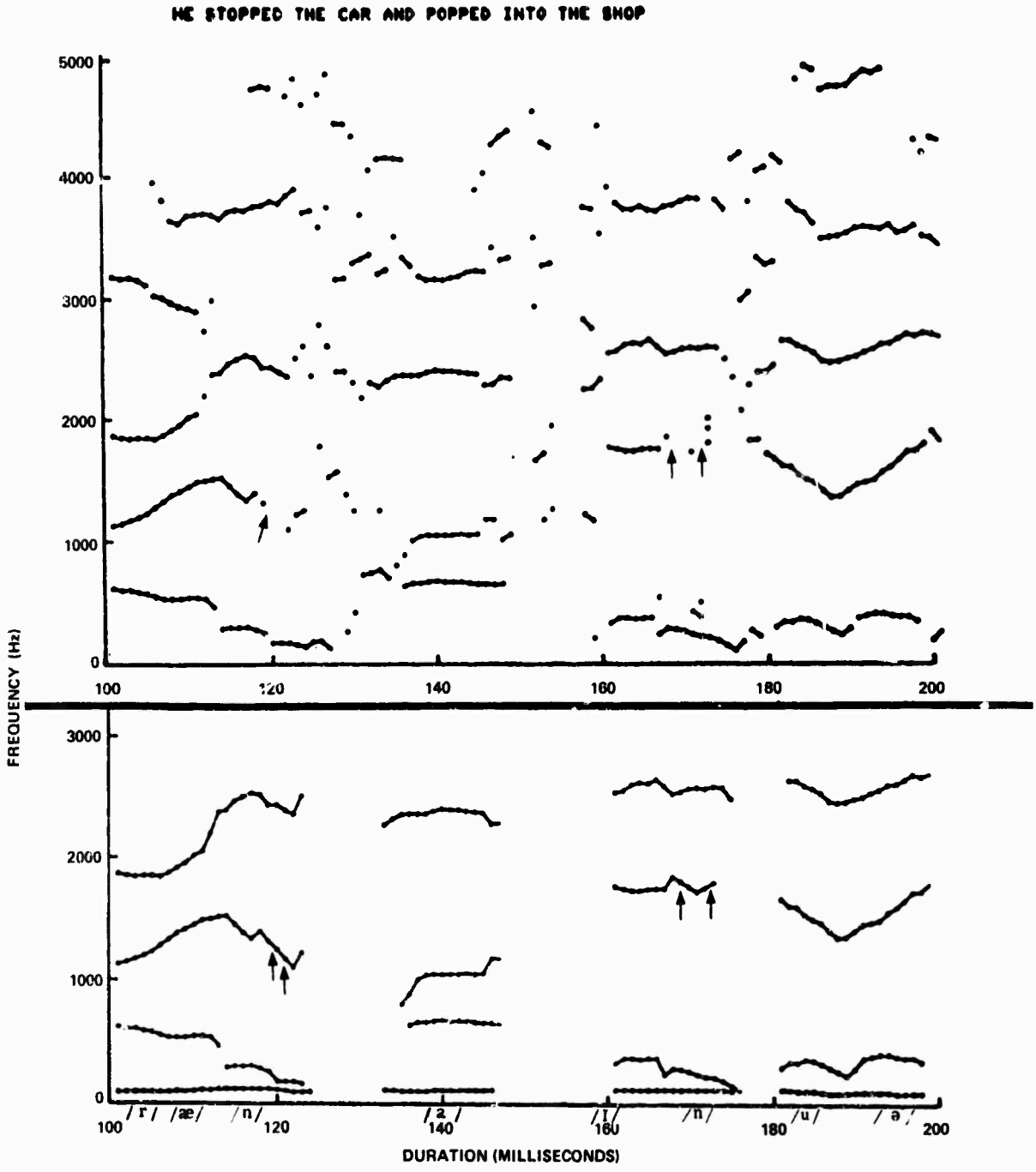


Figure 1-3. Filling in of Missing Formants

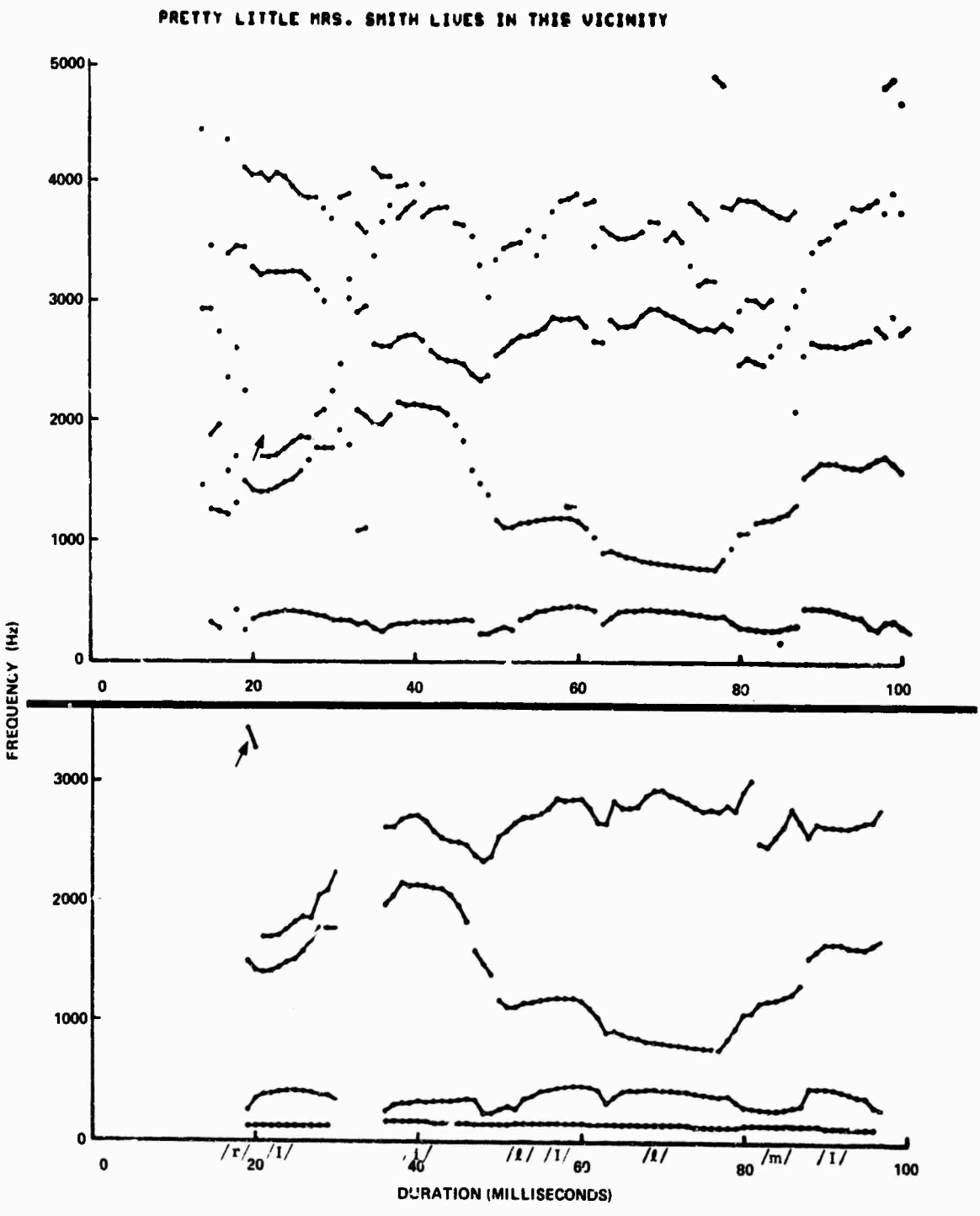


Figure 1-4. Wrong Assignment Due to a Missing Formant

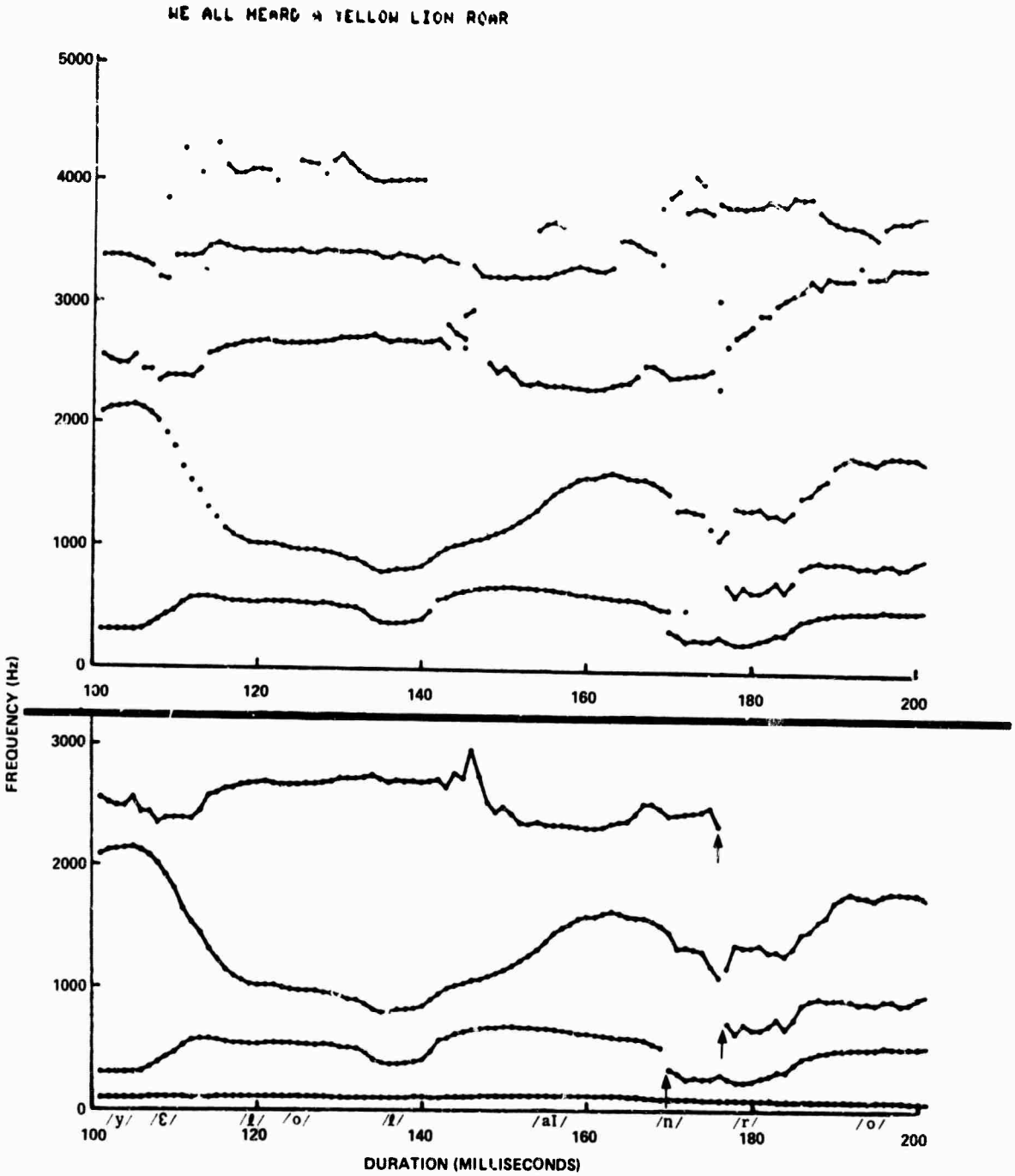


Figure 1-5. Discontinuity at a Nasal-/r/ Juncture

low F1, dropping amplitude in F2, the appearance of a nasal formant that might be confused with F2, or larger bandwidths than occur in the other voiced area.

The main remaining problem areas in formant tracking are /i/, nasalized vowels, and nasals. The problem with /i/ is the occurrence of an extra resonance between F1 and F2. It was found that this extra resonance would be labeled as F2 in the anchor stage, primarily because F2 was weak and had disappeared for one or two segments. A similar problem occurred in nasal areas where there was also an extraneous or nasal resonance between F1 and F2.

Vowel and Sonorant Analysis

The main purpose of the vowel-sonorant (VOWSON) program is to locate steady-state segments and to enter segment boundary and label information into the A-matrix. Not all vowel-sonorant events are steady-state. The definitions of some events are, indeed, tied to the pattern movement of the formant frequencies; they make a gesture toward a target but do not attain the target or do not hold a fixed position for even a short period of time. Also, some events do attain a steady-state, but the target they reach has been influenced by surrounding sounds and does not match closely to "pure" vowel or sonorant targets as indicated in the speaker-dependent tables. The results of a retroflexion experiment indicate algorithms for handling retroflexed vowels, but nasalized and lateralized vowels cannot be meaningfully handled until the appropriate experiments have been performed and the results interpreted.

The strategy of the present VOWSON program is to locate, segment, and label appropriately only those steady-state areas that it can handle with a high degree of reliability. All other voiced areas are left for the lexical mapping procedures to interpret in a syllable, word, or phrase context (see Section 1.3.4, below). VOWSON does provide the mappers with information extracted from the parameters to enable them to map more efficiently. This information is provided in the form of the following kinds of indicators:

- Indications are made in the A-matrix as to discontinuities in F1, F2, or F3 based on the difference in formant frequencies between adjacent frames. Frames having the pattern of an existing F0, F2, and F3 but a missing F1 are indicated as semi-voiced frames. These types of areas tend to be /h/ or /ᴹ/ sounds or whispered trailoff vowels.

- Slope changes in F2 are indicated, along with the direction of the change (either falling or rising). The appropriate rise or fall indicator is turned on for each frame in which the frequency change exceeds a threshold from one frame to the next. This enables the mappers to quickly discern slow-moving F2 changes from those that are moving more rapidly.

- A sporadic voicing indicator is turned on if the fundamental frequency goes on and off over a contiguous period. This indicator is used as a flag to the fricative-plosive program to investigate the area.

- A retroflexion indicator is turned on for all frames in which F3 is below a threshold value. The threshold value is defined as being half the F3 distance between /ɜ/ and /u/.

- A lateralization indicator is turned on for all frames in which the F1, F2, F3 frequency pattern is within a threshold difference of the pattern given for /l/ in the speaker-dependent table.

- A nasalization indicator is turned on when the F1 frequency is low and the F1, F2, F3 frequency pattern is not /i/-like or /u/-like.

- Contiguous voiced areas not exceeding three frames for which formants are missing or erratic are labeled "voiced junk." They may be non-speech phenomena such as tongue clicks, glottal sounds, or portions of bursts.

- A falsetto indicator is turned on for frames having an F0 greater than 350 Hz. A vocal fry* indicator is turned on for frames having an F0 less than 65 Hz.

VOWSON also detects energy dips in voiced areas and indicates the dip areas in the A-matrix. The parameter used for dip detection is the RMS after a three-point average smoothing has been performed. The technique used is similar to that described by Weinstein et al. [24]. Each minimum is tested against its surrounding maxima to ascertain that the ratio of the minimum to each surrounding maximum is within a threshold of .80, and that the combined ratios are within the threshold 1.2. The dip-location technique was applied to 69 utterances from a protocol gathering. Some sample results are given below:

*"Vocal fry" refers to what are often called "creaky voice" sounds.

	Word or phrase (phoneme or boundary underlined)	# Times phoneme (or boundary) found correctly	#Occurrences of word (or phrase)
Detection of voiced plosives (/b/,/d/,/g/)	submarine(s)	18	21
	number	10	11
	submerged	7	11
	Albacore	1	1
	guided	3	3
	guided	1	3
Detection of unvoiced plosives (/p/,/t/,/k/)	what is	3	10
	Washington	3	9
	thirty	3	4
	computer	1	1
	subset on	1	1
Detection of morph boundaries	missile | launchers	3	14
	the | Ethan	2	9

Some other sounds labeled as dips were: of the Soviet, Lafayette, length, united, many.

VOWSON utilizes previously constructed speaker-dependent vowel-sonorant tables. These tables contain entries for the following ARPABET symbols: IY, IH, EH, AE, AA, AH, AO, OW, UH, UW, AX, ER, L, W. Each sound has F1, F2, and F3 frequency values associated with it. The frequency values for IX and R are assigned by the program from existing sounds in the table. The F1 of IX is defined as half the distance between the F1s of IY and IH; the F2 and F3 of IX are defined as half the distance between the respective F2 and F3 values of IH and AX. The F1 of R is defined as 3/4 the F1 value of ER; the F3 of R is defined as the F2 of ER, and the F2 of R is defined as the F1 of R plus 60% of the distance between the F3 and the F1 of R. VOWSON also assigns frequency values to a group of retroflexed vowels: IY, EH, AH, OW, UW, ER. The algorithms used are those described in the results of a retroflexion experiment [7].

The F1, F2, F3 frequency values of the vowels in the speaker table are converted to a linear scale from 0-99. This allows matching to be done on the basis of linear distance rather than Euclidean distance, reducing computational costs. The conversion is made by finding the minimum and maximum F1, F2, F3 values

from the table (excluding the retroflexed vowels) and extending these minima and maxima ±15%. The distances between the minima and maxima are then divided by 99 to yield the scale factor for each formant. Each frequency resonance can then be converted by subtracting the minima for that resonance and dividing by the respective scale factor. An example table for speaker WAB is shown in Table 1-2. Also shown are the F1, F2, F3 minima and maxima values and their respective scale factors. If a formant frequency is below the minimum frequency, its default setting is 0; if it is above the maximum, its default setting is 99. All F1, F2, F3 values in the A-matrix are converted to scaled values, and the scaled values are stored in the A-matrix as additional information.

The initial algorithms for aggregating frames into segments and labeling the segments were incorporated into VOWSON in March, 1975, and are still in the debugging stage. Current plans call for recording the ten test utterances shown in Table 1-1 for six speakers. These utterances will then be used to analyze program performance. The analysis will undoubtedly result in modifications to the program. The following is a description of how the current segmentation program operates.

The first phase of segmentation is to find nuclei within the utterance. Starting at the beginning of the A-matrix and proceeding to the end, each voiced (V) area is located and labeled. Voiced junk areas are ignored. The nucleus finder is run in all areas having the following characteristics:

1. The entire area is labeled V.
2. Each frame in the area has an F0, F1, F2, and F3.
3. The area does not contain a dip.
4. The area is ≥3 frames.

The first task of the nucleus finder is to locate the frame(s) of peak RMS energy in the defined V area. This is done by using the first-difference values between adjacent smoothed RMS values. (There may be more than one RMS peak if the area includes more than a single vowel surrounded by sonorants.) The other parameter used for nucleus finding is the absolute first difference in scaled F1, F2, F3 in adjacent frames for the defined area. If this value for all frames exceeds a threshold, then there is no nucleus, and segmentation and labeling are not attempted in that area. This is because the formant frequencies are moving too rapidly to define a steady-state area, and the problem of how to interpret the area is deferred to the mappers. If there are first-difference values between the threshold, the frame showing the smallest difference (least amount of change) is selected as the nucleus. If more than one frame has the same minimal difference, the frame closest to an RMS peak is selected as the nucleus.

Table 1-2. Vowel-Sonorant Table for WAB

Phone		Hz.			0-99 Scaling	
	F1	F2	F3	F1	F2	F3
IY	273	2304	2851	8	92	81
IH	429	1914	2695	39	64	73
EH	526	1835	2598	58	60	68
AE	625	1660	2343	78	52	55
AA	645	1093	2440	82	25	60
AH	585	1406	2539	70	40	65
AO	645	1054	2617	82	23	69
OW	507	1093	2382	54	25	57
UH	429	1210	2382	39	30	57
UW	351	1152	2246	23	27	50
AX	546	1367	2304	62	38	53
ER	400	1445	1640	33	41	20
L	351	859	2695	23	14	73
W	351	664	2226	23	4	49
IX	351	1640	2499	23	51	63
R	300	986	1445	13	20	10
IY	351	2109		23	73	
EH	462	1952		46	65	
AH	695	1013		82	24	
OW	457	993		44	20	
UW	390	1181		31	28	
ER	414	1679		36	52	

Retro-flexed Vowels

Formant	Minimum	Maximum	Scal. Factor
F1	233	741	5
F2	565	2649	21
F3	1229	3278	20

Once the nucleus is defined, the segment boundaries are determined by moving to the right and left of the nucleus until a scaled F1, F2, or F3 value differs from that of the nucleus by more than a threshold value, or until the beginning or end of an adjacent segment is encountered. More than one segment may be defined in the area if the undefined gaps between segments and/or the beginning and end of the area are greater than a threshold number of frames. The beginning, end, and nucleus indicators for each segment are entered in the A-matrix.

Labeling is done on the basis of the scaled F1, F2, F3 values found in the nucleus frame. Linear distances are computed from each vowel and sonorant in the speaker table; these distances are ordered, and the first four choices (closest matches) below 50 are selected and entered in the A-matrix. The score, at the present time, is simply the linear distance of the match.

Labeling proceeds as follows. If a segment is immediately preceded within six frames by two consecutive frames that have retroflexion indicators turned on, then the retroflexed IY from the speaker's vowel table is used instead of the non-retroflexed IY, and only the F1 and F2 distances are measured for the other sounds. Likewise, if the segment is followed within six frames by two consecutive frames that have retroflexion indicators turned on, then the retroflexed vowel table replaces the non-retroflexed vowel table. If the nasal indicator is turned on for the segment, then a NA (nasal) choice with a default distance of 50 is inserted as the last possible choice in the A-matrix. The default is used because the locations or effects of nasal formants and zeroes are not known at present. No special handling of vowel lateralization or nasalization is attempted at this time.

The nucleus finder, segmenter, and labeler are also run on dips if they exceed a threshold number of frames and all frames have an F0, F1, F2, and F3. If the dip is short, a default nucleus is defined to be the middle frame.

Lateralization Studies

An experiment has been designed to compare the formant frequency spaces of lateralized versus non-lateralized vowels, as well as the characteristics of initial, final, and syllabic /l/ and the behavior of /l/ in consonant clusters. Results from this experiment will be incorporated into the vowel-sonorant program for the Milestone System.

Fricative/Plosive Analysis

Previous research [13] has produced techniques for handling unvoiced fricatives and plosives that have been very effective in building an accurate A-matrix of an utterance. These techniques are being extended to allow analysis of voiced fricatives and plosives, and are being significantly improved by the utilization of accurate formant and voicing information. Additionally, analysis data are being stored in the A-matrix to assist the lexical mapping procedures.

Performance analysis of the Voice-controlled Data Management System (VDMS) of December, 1973, demonstrated basic strength in the fricative/plosive analysis being performed, in that the linear prediction correlation (LPC) technique we had developed (the "low-coefficient" LPC, or LCLPC) was quite accurate in its modeling of unvoiced burst and frication spectra. But a number of weaknesses became apparent:

1. The limitation of the LCLPC to unvoiced speech was undesirable, and not merely because we lacked results for voiced fricatives and plosives. The LCLPC was, unfortunately, sensitive to the presence of voicing, even small amounts of which led to mislabeling. Worse, because the A-matrix lacked an accurate voicing measure, the fricative/plosive program often analyzed utterance areas that could yield no answers at best, or wrong answers at worst.

2. The fricative/plosive program could not incorporate well-known acoustic-phonetic findings regarding formant trasitions, since formants were not tracked.

3. The fricative/plosive program did not fully exploit the capabilities of LCLPC analysis. This was most apparent in that no attempt was being made to track spectral change in the frication and burst portions of an utterance despite the availability of consistent and reliable LCLPC information.

4. It became obvious that speaker-dependent parameters were as necessary to fricative/plosive analysis as to voiced analysis, where we had successfully applied them. The fricative/plosive program needed to incorporate speaker dependency.

5. Insufficient intermediate fricative and plosive analysis data were available in the A-matrix for the lexical matching program to optimally perform either a quick, approximate mapping or an accurate remapping; all information was at the same level of detail.

Major progress has been made in removing all of these weaknesses in the new fricative/plosive program being developed for the Milestone System.

1. The reason why LCLPC analysis yielded useless spectra from voiced fricatives and plosives was that the linear prediction algorithm would assign poles to modeling the uninteresting but high-amplitude, low-frequency "voicebar" area of the spectrum. We explored several ways of removing the influence of low-frequency energy on the LCLPC spectrum without destroying its performance as a burst/frication model. A carefully selected finite-impulse-response digital highpass filter proved to be quite effective over a number of utterances. The filter was recoded in fixed-point assembly language and embedded in the existing integer LCLPC program, and was successfully retested in February. Burst and frication spectra in voiced and unvoiced fricatives and plosives are now analyzable with the same tool.

2. Availability in the A-matrix of information derived by automatic formant tracking now allows formant trasition analysis. An experiment is planned for later this year to verify that voiced-plosive place distinctions may be reliably made on the basis of preceding and following formant information, as has been previously reported (Öhman [16]). Quantitative exploration of speaker dependencies will be made at the same time.

3. Spectral change information will be computed from pairs of LCLPC spectra, with numeric results stored in the A-matrix. The routine to do this has been coded as a part of the new fricative/plosive program.

4. The need to incorporate speaker-dependent information was taken into account during the initial design of the new fricative/plosive program; both speaker-normalized and unnormalized information will be present in the A-matrix.

5. Intermediate inform tion useful to the lexical mapping procedures was specified during redefinition of the A-matrix in October, 1974, and is being incorporated during the coding of the fricative/plosive program.

The initial version of the new fricative/plosive program is presently being coded. It is expected to be operational in June.

1.2.2 Lexical Mapping

The system developed jointly by SDC and SRI uses the same basic acoustic-phonetic processing and lexical mapping techniques that were used in the VDMS system. The lexical mapping routines (i.e., the MAPPER program) were improved and expanded to provide the necessary support for the best-first parser. The vocabulary was expanded, and some new rules were included. Some new capabilities, such as lexical subsetting, were developed and are described below.

While we have been adapting MAPPER for use in checking out the joint system, we have gone ahead with the design and implementation of a new set of lexical mapping modules that will be part of the 1975 Milestone System. They are being designed specifically to interface well with the best-first parser, to take advantage of the greater space and flexibility being provided by the CRISP system, and to utilize the results of the greatly improved acoustic-phonetic information found in the A-matrix that will be used in the Milestone System.

MAPPER Development

During the time in which the SRI best-first parser was being implemented in LISP/370, the MAPPER program was extensively tested, modified, and evaluated. Errors were fixed, algorithms were improved, and techniques on all levels were examined. The principal testbed for MAPPER was VDMS. Although the data management language was not normal English, it was in several respects nearly ideal for exercising MAPPER. All "function words" in the language carried an important functional load, and because they were comparatively few in number, they served a useful purpose in directing the VDMS parsing modules along the more probable paths. This allowed primary attention to be directed toward MAPPER rather than the parsing modules. On the other hand, the language contained constructs in which the normal fan-out was 60 to 65 "content words," out of a 150 word vocabulary, and MAPPER had to try these to attempt to choose a best one.

Intrinsic non-directionality in VDMS forced attention to the more difficult problems of word boundaries in mapping. A word could be predicted adjacent to a word previously mapped or in a syntactic "hole," in which no specific boundaries were known. In the latter case, MAPPER had to search through the region it was given, looking for a plausible place to locate itself to begin actual mapping.

Using the Control Structure Language (CSL), another parser for the data management language was built--a "mini-best-first" parser. Its implementation was comparatively easy. It achieved approximately the same accuracy on the utterance level as the VDMS parser.

A corpus of 92 utterances was used to test MAPPER, 62 by one male speaker and 30 by another. The testing at this stage had as its purpose obtaining the best possible mapping from the existing acoustic-phonetic processing. Seven changes and improvements resulted from this testing.

(1) Various kinds of trade-offs were evaluated.

(2) The lexical base forms and the phonological rule set are tuned to each other in the sense that a choice in one affects the other; as a result of the testing, some base forms were revised and some new rules were added.

(3) Low-level phonetic routines that perform pattern matching in the A-matrix were examined to determine more nearly optimal constraints on deviation from ideal patterns.

(4) Scoring on all levels was scrutinized to determine when the results of a mapping should be accepted and when they should be rejected. Ad-hoc scoring algorithms were juggled, and in some cases a more empirical basis was arrived at for calculating scores.

(5) The handling of predicted word boundaries was made more sophisticated so that gaps and overlaps could be handled more satisfactorily.

(6) In some instances, a correct word might have been missed because a part failed to map correctly; a capability was added to attempt provisional remapping for parts of a longer word where at least half of its syllables had mapped successfully.

(7) Finally, certain implementation techniques were somewhat revised: an example was that a recursive sequence of functions for mapping syllables to build words was changed to an iterative sequence in order to provide simpler control over certain aspects of the syllable mapping.

The testing phase also provided the opportunity to evaluate MAPPER from various other points of view. One perspective was its role as a basic acoustic-to-lexical mapper for the integrated SDC-SRI system. A second related to pin-pointing consistent weaknesses in the A-matrix information, and the third was scrutiny of our approach to lexical mapping and the manner in which it was implemented.

Because mapping is a comparatively time-consuming process, it is prudent to avoid needless replication of processing. The early VDMS contained a specific storage technique for information about words already correctly mapped. It was not convenient to use this same storage technique with the joint system; a new facility, called the Lookaside memory, was therefore built directly into MAPPER. The Lookaside memory is basically a dual array, each part having an element corresponding to each 10-ms. frame in the A-matrix representation of an utterance. The Lookaside memory is updated after each call on MAPPER; that is, the parameters with which MAPPER is called and the result of that call are stored away. If a subsequent call to MAPPER is made with the same or very similar parameters, the same answer as before may be returned without remapping.

The Lookaside memory contains both positive and negative results, the negative ones being indicated by a null score. The necessity for a dual array arises from the fact that MAPPER is most commonly called with either a known left boundary (i.e., a left-to-right mapping) or a known right boundary (i.e., a right-to-left mapping), but not both. Therefore, when a word is not found, it is known to not exist only either to the right or to the left of some point. This implied directionality is expressed by using the two arrays: a forward (left-to-right) lookaside and a backward (right-to-left) lookaside.

The Lookaside memory was tested under VDMS. On the longer utterances (four or more words), as much as 50% of mapping effort was saved. (It was of little help on very short utterances.)

A secondary result of testing MAPPER was the collection of information that is being utilized in the design and implementation of new acoustic processing procedures for building the Milestone System A-matrix. This information was of two kinds; one was the correction of deficiencies in the old A-matrix, and the other was knowledge about acoustic-phonetic tendencies and regularities that could be moved out of MAPPER and into the building of the A-matrix itself. It was observed, for example, that use of isolated spectra for assigning values to the first three formant frequencies resulted in fairly frequent formant errors and, therefore, errors in vowel recognition. A reliable formant tracking program will minimize this type of error in the Milestone System. Errors involving voiced fricatives and plosives have led to development of processing techniques that provide greater identification and discrimination of these phones.

Coarticulation of various phonetic segments with adjacent vowels was handled in a primitive way in the low-level vowel-matching routines of MAPPER. During testing, more regularities were observed, and this led to research efforts, such as the retroflexed vowel experiment, to classify the regularities and to write algorithms to describe them. This information is now used in the vowel-labeling procedures.

Finally, the MAPPER testing provided an extensive evaluation of both the overall approach taken by SDC in top-down mapping, or analysis by synthesis, and the techniques employed to implement that approach. Analysis by synthesis as a verification procedure performed well in the context of a system that hypothesizes words, the language being constrained by syntax, semantics, and pragmatics. MAPPER had a high degree of success in picking out the correct word from a list of predictions. In the Milestone System, the language is much less restrained, and supplemental mapping techniques will be required. Top-down mapping, however, will remain as an important procedure in the speech understanding strategy.

The MAPPER developed within VDMS made explicit use of the syllable as a phonological and phonetic unit. Syllable boundaries were marked in the lexical base forms and used in the phonological rules. At the phonetic level, the syllable was represented by a data structure having a vowel nucleus and possibly consonant clusters on either side. This framework was used successfully to deal with coarticulation phenomena such as pre-stressed plosive-sonorant sequences and nasalized vowels. Testing indicated the specific instances in which the algorithms had difficulty in locating a syllable nucleus or its boundaries. This knowledge will be used to make the new MAPPER more versatile, so as to exploit the syllable approach to an even greater degree.

Integration with SRI Parser

A first step in integrating MAPPER with the SRI parser was separating it from VDMS. Interconnections were severed, files of MAPPER code were grouped

together, and modifications were made to the calling sequence so that MAPPER could easily be called as a function. Planning for the joint system resolved issues of system design and defined the interface between the parser and MAPPER. It was recognized that various kinds of supplemental mapping capabilities would be required, including capabilities for lexical subsetting, prosodic phrase mapping, and, eventually, acoustic bottom driving. However, the initial mapping was to be performed by the existing MAPPER, enhanced to function well with the new language.

Preparing the MAPPER for joint system integration involved a number of tasks. The English subset used contains words with inflectional endings, and the phonological rules system includes these affixes in its predictions. Thus the word "submarines" consists of the syntactic elements "submarine" and PLURAL, and the number "seventieth" consists of "seven," TEN, and ORDINAL. Affixes have different forms, depending upon the phonological conditioning of the roots they are attached to. Roots also may change when affixes are added. Derivational rules are required to generate the correct phonemic string from a given root and its affixes. Incorporating the derivational facility into MAPPER required generalizing the mechanism for constructing phonetic spellings and expanding the phonological rules system [1,2].

An early version of a phonological rules system was developed for generating variant pronunciations of lexical entries. It assumed that rules were applied in an unordered, optional manner. A different set of assumptions is required for rule sets whose task is to derive inflectional or morphological endings. These rules are ordered and obligatory (if the context criteria are met), and successive rules operate on the output of the preceding ones so that only one spelling is derived. (These are the types of rules more often discussed in the linguistic literature.)

The phonological rules system has been expanded to a much more generalized facility. It now provides for the building of lexicons and sub-lexicons. A lexical item may be tested individually or as part of a sub-lexicon. In this system there are three types of rule-driver subroutines: ordered, unordered, and nondeterministic. Unordered and nondeterministic rule applications are very similar, the only difference lying in the fact that in a small number of cases, a rule that would apply after a previous rule in a nondeterministic case would not apply in the unordered case because its left context was altered by the previous rule. The phonological rules system was designed as an independent rule-evaluation program. It has been slightly modified and incorporated into MAPPER.

Lexical base-form spellings exist as properties of the orthographic words in a specially coded array structure. The phonological rules system makes use of this array coding during rule application; the result is that new coded arrays corresponding to variant spellings are produced. Under the old technique, the orthographic word was predicted, and its base-form property

was extracted in MAPPER. When a word can be predicted with one or more affixes, then the old approach is not adequate; the entire phonetic string must be derived and mapped as a whole. Routines were developed to construct new coded spelling arrays by copying one or more old ones. MAPPER now receives as one of its input parameters a list of one or more words and/or suffixes. The spelling of each word is extracted; if a word has suffixes, they are derived using the ordered rule driver. The result is a single coded array, which may then be passed to the unordered rule driver for generating alternative pronunciations and mapping each one of them. This allows whole phrases to be mapped, with the added advantages that variants may be generated that result from applying coarticulation rules across word boundaries that are internal to the phrase.

MAPPER was integrated with SRI's best-first parser in an orderly manner. Because of LISP/370 address-space considerations, MAPPER and the parser were compiled into separate copies of LISP and communicated with each other through a "co-module" package, which contains a parameter-passing mechanism. For initial checkout of the joint system, a small corpus of ten utterances, based on earlier protocol experiments with the submarine data base, was used. A lexicon was built for these utterances, and inflectional rules for deriving plurals and -ty and -teen suffixes for numbers were provided. The first utterance was successfully recognized on the first day of a two-day integration period. More recently, the entire 300-word lexicon has been implemented, and suffix rules have been written to derive all of the 13 kinds of suffixes contained in the full grammer.

Lexical Subsetting

Because the joint system has a comparatively rich grammar, allowing a wide range of English constructions, it is difficult to rely completely on predictions from the parser, particularly in beginning a parse. A lexical subsetting capability was planned early in the period of joint effort, but because of time constraints, its actual implementation was delayed until after the joint system was operational. Lexical subsetting was simulated by a routine, written within MAPPER, that predicted the entire lexicon at any given time boundary and returned a list of elements in order of their mapping scores.

After system integration and some preliminary checkout, work was begun on real lexical subsetting routines. There were three principal design constraints:

(1) they should be fast,

(2) they should rarely reject a correct word, and

(3) they should make maximal use of the acoustic-phonetic information found in the A-matrix.

Since the number of lexical entries upon which subsetting takes place could be quite large, extending to the entire lexicon in some cases, it was determined that speed could best be achieved by using a strategy based on acoustic-phonetic information. An analysis of the A-matrix information would be done only once each time the subsetter was called. The A-matrix pattern would then be assigned to one or more classes. Other routines would subset the input word list on the basis of whether or not a given word manifests one of the acoustic pattern classifications. The second criterion is met by allowing for impreciseness and errors in the A-matrix information. There are a few "catch-all" classes that include phones whose acoustic correlates are ambiguous or nonrobust. On the other hand, certain cues in the A-matrix are very reliable. When they are present, words having their class designations are placed high in the list of candidate items; this satisfies the third criterion.

There are actually two subsetters, one that moves right from a given input boundary and one that moves left. In subsetting, it is also necessary to deal with the possibility of overlapping phones (e.g., The Ethan Allen). The subsetters take care of this problem by returning two lists of items--one from an analysis beginning at the time frame specified as the input parameter and the other from a time five frames (50 ms.) preceding the input frame for the right subsetter or five frames following the input frame for the left subsetter.

In addition to the functions used when a subsetter is called, there is also a set of preprocessing functions that examine the phonemic spelling of each lexical entry and assign that entry to one or more subsetting classes. These preprocessing functions run when the lexicon is built.

The following is a brief description of the subsetter technique. (The right subsetter is used for purposes of illustration; the left subsetter works essentially in the same manner except in the opposite direction. The right subsetter gives somewhat better results because (1) phonological constraints are tighter on word beginnings and (2) prevocalic consonant cues tend to be more robust.) From the time frame specified by the input parameter, a routine moves right in the A-matrix looking for the first vowel-sonorant-like area. It sets pointers to the beginning and end of this area. It also indicates the possibility of a sonorant's occurring to the left of the vowel. Using these pointers, another routine attempts to locate the vowel nucleus based on energy considerations. If more than one energy peak remains after smoothing, certain criteria are invoked to choose the (hopefully) best one. The vowel nucleus, once obtained, provides the anchor point for further analysis. When one works back from the vowel, the phonological constraints of English greatly limit the possible phone sequences that may begin a word.

Analysis is continued from one of three routines--one for sonorants, one for nasals, and one for vowels. The region between the input boundary frame and the beginning of the voicing of the vowel-sonorant area is examined in the

A-matrix for various patterns of phone sequences. All plausible patterns are returned by these functions. A function takes the list of patterns found and maps them into a smaller set of classes. This mapping also results in ordering of more specific patterns in lists before more general ones. For example, patterns having an L or W in them get higher priority than one indicating that "some sonorant" may have been found. At this point the analysis and classification have been completed.

What remains in the subsetting procedure is primarily the shuffling of lists. The first input parameter has to do with the set of words being processed. If the parameter is ALL, then the subsetting operation is performed on the entire vocabulary. Otherwise, the parameter passed is a smaller subset of candidate words. In the preprocessing phase, the vocabulary is divided into right subsetter classes. If ALL is specified, then these classes are joined together in a list of lists, maintaining the orderings as previously given. If a smaller candidate list is input, the class properties of each word in the list are examined for a match. When one is found, that word is added to a list, and lists of lists are formed as before.

One final pass is made on the resulting list to further eliminate extraneous words. Vowel quality and crude stress criteria are applied, using the previously found vowel nucleus. This particular part of the procedure has worked very well; of the five vowel classes that are defined, the first vowel of the correct word is almost always in the correct vowel class. The resulting list, together with the time frame boundary, is passed back to the calling function. It should be remembered that another very similar calculation, but possibly with considerably different results, is also performed at a time frame 50 ms. earlier, to allow for overlap. Obviously, the calling function may use as much or as little of the returned information as is useful.

This version of the subsetter routines is based on the old A-matrix information and is limited in its capability by that information. The acoustic-phonetic processing for the new A-matrix is much more sophisticated, and new subsetters based on it can likewise be expected to be more selective and more accurate. Experience gained in the two versions of the subsetters can be expected to provide a great deal of insight for an eventual full-blown bottom-driving capability based on phonetic syllabic analysis.

Preparations for the Milestone System

In the course of developing the 1975 Milestone System, we have increased the number of processes that may mediate between the acoustic-phonetic data and the parser. What was once a single module, the MAPPER program, used in top-down word and phrase verification, is being replaced by several new modules. First of all, a new top-down mapping procedure that will have the same basic function as the old MAPPER is being developed. Improved versions of the lexical subsetting modules will also be developed. A prosodic phrase mapping capability that takes advantage of prosodic analyses will be provided for the parser to use in verifying the prosodic plausibility of found word strings. Finally, a

limited, syllable-based acoustic bottom-driving module will propose words to the parser to help get the parsing started and to open up analyses at new points in an utterance.

The philosophy and techniques of the lexical subsetting modules described above will continue to be used in the new versions. Prosodic analysis is described below in Section 1.6. What follows here is a discussion of topics related to the new top-down mapping routines and then some considerations in the design of the "word spotter," or acoustic bottom-driving module. Some issues that result from theoretical questions and others arising out of system evaluation are also discussed.

The new top-down mapping module is called MAPPER.2. CRISP, which will allow the entire system to be written as a single program module, will also provide a number of advantages for MAPPER.2. Some of them are:

(1) greater versatility and generality of data structures;

(2) language capabilities for efficient and fast arithmetic operations;

(3) greater versatility of programming techniques; and

(4) sophisticated control mechanisms for functions and processes.

Complex kinds of data structures are found in various parts of the mapping routines; they greatly facilitate the necessary "housekeeping" tasks. MAPPER.2 will profit from the accessibility of n-node lists and n-tuple structures that were not available in LISP. Much of the mapping code uses language facilities and programming techniques that are more representative of procedural and algorithmic languages than of the pure list processing and recursive functions that are characteristic uses of LISP; CRISP combines all of these capabilities into one language. Because of the desirability of extending the "best-first" concept even into various levels of MAPPER.2, the flexible process initiation, suspension, and termination control mechanism provided by the language will prove highly useful in the implementation. The parser will have the ability to suspend, at least temporarily, a mapping that appears to be producing a less profitable result than some other activity that is also in progress.

A primary objective in the design and implementation of MAPPER.2 is to bring mapping techniques abreast of the latest technology in acoustic processing at SDC. While top-down mapping will remain much the same in philosophy, it will be more powerful and more efficient than the old implementation. Pronunciation variants will be generated directly into the lexicon and will be mapped in parallel rather than sequentially. Rule application will thus cause a linear, rather than exponential, increase in mapping time. Top-level procedures in MAPPER.2 will have access to more global information about the

mapping of a word; syllable mapping can thus be optimized. For example, in the case of a vowel/voiced-consonant/vowel sequence, the two syllables will be mapped together, avoiding the difficulties that arose in the old MAPPER.

MAPPER.2 will have considerably more information about where to look in the A-matrix; in fact, the A-matrix itself will provide this information. One result of our recent work in acoustic-prosodic analysis is the location and (partial) isolation of phonetic syllables. Combined with this is some indication of vowel stress, and information given by the vowel-sonorant program about the location of vowel nuclei. Being able to locate precisely the region in which a hypothesized string is to be mapped will enable MAPPER.2 to decide quickly whether the string fits at all; if it does fit, the actual mapping will occur in the right place. Having available the location of most syllables, MAPPER.2 will also be able to make use of rate-of-speech information. The vowel vs. consonant time, which has been much discussed in the acoustic-phonetic literature, will finally become a meaningful and useful item of information in a speech system.

Low-level mapping routines that access the A-matrix will be completely new in MAPPER.2. They will make use of formant trajectories, segment boundary marks, and reliable voicing information, as well as vastly improved segmental phone classifications. Phone sequences such as /ks/, /ts/, /ps/, /kl/, and /tr/ often function as units phonetically, and they will usually be so treated by the MAPPER.2 search routines.

Development of the joint SDC-SRI system has already shown that considerable care must be taken to provide all of the information that is necessary at the interface of two major modules. MAPPER.2 will be designed to a considerable extent around the parser's needs and capabilities. Already it has become obvious that as a grammar and a phonological component both increase in complexity, the boundary between syntax and phonology becomes more and more fuzzy. For example, regular plurals of nouns are derived by rule, while irregular pluralized nouns occur as separate entries in the lexicon (e.g., foot, feet). The grammar has irregular pluralized nouns in a separate class so as not to attempt to derive the plural of an already plural noun. The derivation of past-tense and past-participle forms will require a more elegant solution; each verb will require marking in the lexicon with respect to the kind of derivation used to obtain the past-tense and past-participle forms.

A parsing strategy for speech has imposed upon it a number of constraints not present in a teletype-input system; for example, a strict left-to-right parse is often not possible. A particularly difficult problem that arises is that the parser cannot assume that all syntactic terminal elements can be treated alike. Some syntactic terminals (e.g., various suffixes) have little or no independent representation in the acoustic information; others (e.g., function words such as "a," "the," "of," "an," "and") are similar to each other acoustically and also are so short as to have a very high false-alarm rate because parts of so many other words contain something that looks like them. An important activity in preparing for both the Milestone and the five-year

systems is investigating carefully the instances in which the parsing stragegy must compensate for inadequacies in the acoustic data. This means modification of some of the parsing strategies, and perhaps, in some cases, slight modifications in the philosophy of the parser. Arriving at satisfactory solutions will require careful coordination between the two sites responsible for the system.

In many respects, the lexicon is the heart of any natural-language system. It must contain various kinds of syntactic, semantic, and morphological information. When the input to the system is speech, phonemic and/or phonetic representations must be included. In the joint system, the syntactic and phonological lexicons were separated because they were used in program modules that were separate copies of LISP. This was feasible because, using the small test vocabulary, the dependencies between the two types of lexical information were minimal. With the use of a full syntax and larger vocabularies, interdependencies will proliferate, and it will be preferable to make the lexicon a unified data structure or set of structures. This is another of the many areas requiring careful planning and design on the part of personnel from both sites.

A specialist contractor, Speech Communications Research Laboratory (SCRL), has also been actively assisting in the development of lexicons for the system. They have provided support in helping to develop base forms to be used for lexical entries. They have also been active in the related task of defining and evaluating rules for generating pronunciation variants from the base form. For part of this task they have used our phonological rules system. The 300-word submarine data base lexicon has been implemented, and SCRL is now assisting with the expansion to 600 words. Provisions will be made in the Milestone System for probabilities on rule applications; SCRL may also assist in gathering the necessary statistics to provide meaningful probabilities.

When a system makes use of syllables on its mapping procedures, lexicon development for the system raises significant theoretical issues. It is assumed that there are two related but distinct uses of the syllable concept; there are linguistic or phonological syllables and there are phonetic syllables. The first has to do with convenience in writing phonological rules and the second with physical events during a speech act. Our use of syllables in the old MAPPER implicitly assumed that the correspondence between abstract and phonetic syllables was sufficiently close that the distinction could be ignored for purposes of mapping. It assumed that a syllable either existed or it did not; no provision was made for a "half-way" phonetic syllable, and, in fact, the mapping technique was sufficiently loose that a good score could often be obtained for either of these variants.

Since the lexicon has syllable boundaries marked, it was necessary for the SCRL staff to know what criteria should be invoked to determine where phonological syllables should be divided. In a series of joint meetings, several criteria and algorithms were discussed, and a provisional operational procedure

was adopted for current lexicon development. We expect that continuation of joint studies on syllable division will result in significant contributions to phonological science. At the same time, the fact that the old MAPPER functioned well is a reminder that the construction of useful systems need not be dependent on the solution of all of the theoretical issues.

While a predictive (or top-down) mapping technique may ignore the distinction between phonological and phonetic syllables, an analytic (or bottom-up) technique cannot. The acoustic-phonetic analysis part of such a technique must work with what information it has. The identification of phonological syllables on the basis of analysis and classification of phonetic syllables is a much more difficult step than the reverse procedure. Nevertheless, the syllable approach [11,12] for "word-spotting" has the advantage over other techniques that acoustic cues for syllables are relatively simple, involving primarily energy and (to a lesser extent) pitch and duration.

In conjunction with the very useful information obtainable by the VOWSON program in building the A-matrix, a syllable analysis module will be built using strategies similar to those developed by Mermelstein [11,12] and by Fujimura [5]. Starting with the vowel (syllable nucleus), analysis moves out in both directions to define a syllabic pattern. This pattern is then classified as a syllable type. One or more syllable types may then be looked up in a syllable-keyed lexicon (syllabary) to retrieve the orthographic spellings of words satisfying the syllabic sequence. The list of found words may then be used directly by a higher-level module or may be passed to MAPPER.2 for a tighter map. In addition to developing the syllable analysis routines, effort will be required to work out a suitable inventory of syllable types, to design a suitable data structure for the syllabary part of the lexicon, and to develop efficient lexical retrieval routines. The success of the first lexical subsetting routines indicates that this kind of bottom-up analysis is both viable and attractive.

1.2.3 System Hardware and Software

Computer System

Our Digital Equipment Corporation (DEC) PDP-11/40 computer system, which has been arriving in pieces since June, 1974, is now complete. The configuration is shown in Figure 1-6. All hardware has been installed and is operating, except for interface units, which are ready to be installed as soon as the facility construction (discussed below) is finished. There have been no significant problems with the DEC hardware to date.

Unfortunately, the same cannot be said of the Signal Processing Systems, Incorporated (SPS) SPS-41. Delivery, originally scheduled for April 15, 1974, did not occur until November 27, 1974. The 7-1/2-month delay must not have been for quality assurance; our unit failed upon installation and again two days later. Furthermore, standard SPS support software was

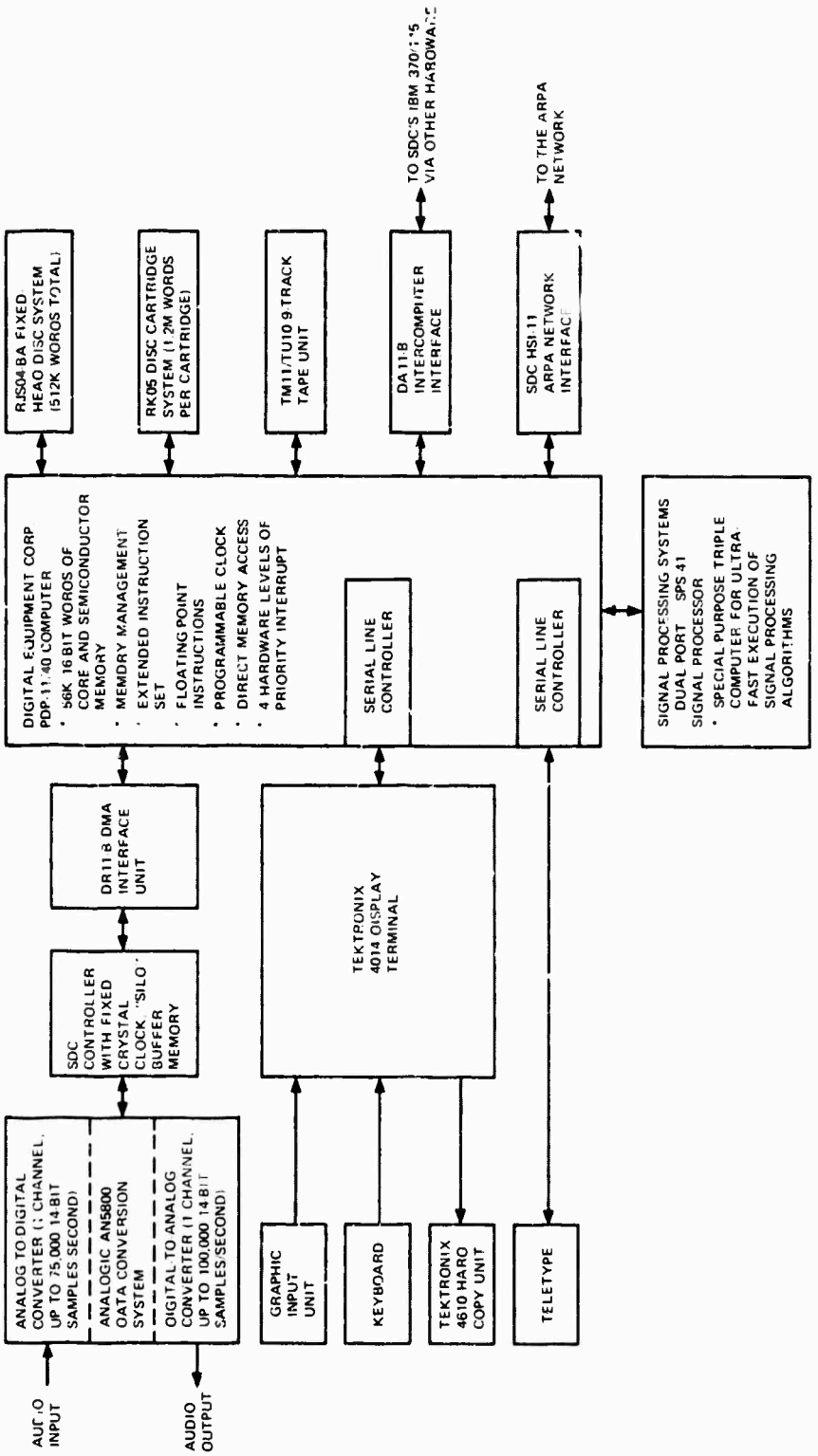


Figure 1-6. PDP-11/40 Computer System Configuration

"delivered" several times in unreadable form, and reports of other ARPA SPS-41 users demonstrated basic design problems that appeared when real application programs (as opposed to SPS diagnostics) were run.

We have been cooperating with other ARPA users of the SPS-41 to verify hardware problems, to work out approaches to their solution, and to impress the vendor in effective ways with the need for solution. Recent reports of helpful SPS engineering modifications have been encouraging, but it is not clear as yet whether these are real and effective corrections. We plan to have our SPS-41 modified, but not until we are convinced that SPS has solved all outstanding problems.

Digital Record/Playback Subsystem

A digital record/playback subsystem is being assembled for our PDP-11/40 computer based on experience with our Raytheon 704 computer system. As on that system, an amplified speech signal is digitized directly in real time with no intervening analog recording and is stored on a fast fixed-head disc. This recording process is reversed for playback. All data are moved using automatic direct-memory-access (DMA) hardware to allow high sampling rates; additional hardware assures unbroken continuity of sampling. The sampling rate is crystal-controlled for high absolute accuracy and long-term stability.

Speech will enter the new system at high quality via an AKG condenser microphone or a Sennheiser headset-mounted microphone. (The latter has interesting convenience and noise-canceling properties that we plan to explore.) The speech signal is amplified using low-noise, low-distortion equipment and is band-limited by a 9,000 Hz low-pass filter (having 40 dB attenuation at 10,000 Hz) before being sampled at 20,000 samples per second.

Our experience with user variability has led us to employ a 14-bit analog-to-digital converson system (Analogic AN5800). Excellent digital recording can thus be obtained without any user gain adjustments. (On our existing system, interspeaker level adjustments via a single gain control typically span about 12 dB, while interutterance peak-level variations for some speakers add another 12 dB. These variations have to be dealt with.) Employment of wide dynamic range in conjunction with a low-noise environment assures good speech input during interactive discourse. The use of analog compression, limiting, or AGC circuits, whose unpredictable dynamic effects would complicate subsequent parameter extraction, is thus avoided.

A standard DEC DR11-B DMA interface provides block-transfer input/output for speech data to the PDP-11/40 Unibus. (In view of hardware problems, our decision not to rely on the SPS-41 for speech input/output seems to have been a fortunate one.) In order to assure continuous sampling during the time required to reinitialize the DR11-B between block transfers, the SDC-designed controller provides a 64-word first-in, first-out buffer. This controller also includes timing and Analogic-to-DEC interface circuits.

Basic design of the digital record/playback subsystem was completed in October, 1974; required parts have been delivered. Detailed design of the audio pre-amplifier/amplifier unit has been completed, and this portion is currently being fabricated, with estimated completion by the end of April. The remainder of the subsystem will be completed and ready for use in July.

Laboratory Facilities

A new physical facility has been designed to our specifications and is presently under construction. It approximately triples our laboratory floor area. Included is an appropriate area for the PDP-11/40 and SPS-41 systems, an area for the IMP and 370/145 interface hardware that is sufficiently close to allow Local Host interfacing of the PDP-11/40 to the ARPANET, and a new IAC sound booth. Plans have been arranged so that the construction process does not disrupt our work. The entire area will be completed and in use by June.

Network Hardware Activities

In late 1973, an ARPA Network interface for the PDP-11 was developed at SDC. Designated the HSI-11A, this interface has been operational at SDC since January, 1974. In March, 1974, SDC was asked by the ARPA Interface Steering Committee (ISC) to submit HSI-11A for possible selection as a standard ARPANET interface for PDP-11 computers. In May, the HSI-11A design was selected. For several months thereafter, ISC members and the SDC staff conducted technical discussions, primarily by ARPA Network Mail, to specify an HSI-11B design suitable for production by some organization for widespread, general use on the ARPA Network. These discussions resulted in 11 engineering changes to the original HSI-11A design in order to meet ISC requirements. A documentation package on the HSI-11B was released in November, 1974 (Molho [14]). Also, SDC has assisted in prototype-building activities at Rand and BBN and has provided consulting services, as required. The prototype design has been forwarded to DEC at ISC request. The result of SDC's effort in this area will be that PDP-11 computers may be interfaced to the ARPA Network using reliable, off-the-shelf hardware.

1.2.4 CRISP

In addition to system hardware and software efforts on the PDP-11/SPS-41 computer configuration, system support work for the SUR effort is being done in the form of the development of a programming language and system that is specifically designed for the implementation of SUR systems. This language, called CRISP, will be operational in July, 1975, on the IBM 370/145 under VM/370. A first draft of the language and system design document was completed in December, 1974 [3]. CRISP offers a set of capabilities that, taken together, make it a uniquely appropriate tool for the implementation of our speech understanding systems. Among these capabilities are:

- A structured data capability similar to that available in PL/1.
- Flexible pointer manipulation similar to that in LISP, including functionals.

- Multi-processing and "spaghetti-stack" primitives.
- Efficient compilation of both arithmetic and pointer-manipulation algorithms (incremental and batch modes).
- Three levels of extensible languages available to the user:
  (1) Source Language (SL)--an ALGOL-like language with infix operators.
  (2) Internal Language (IL)--a LISP-like Polish prefix list structure language.
  (3) Assembly Language (CAP)--a macro-assembly language.
- Availability of dynamic, local, and own variables.
- Name pooling.
- System aids to better utilize virtual memory resources.
- A variety of aids for group construction of large programs.

Also being prepared is a translator program that converts SDC Infix LISP to CRISP/SL.

Using CRISP, the bottom-end numerical algorithms, mapping procedures, and top-end component may all be combined in a single language and system without loss of efficiency. This is advantageous for several reasons, the most important being the increased ability of the modules to coordinate and communicate with one another.

1.2.5 Protocol Experiments

The conduct of protocol experiments represents an important aspect of our system-building strategy. The dialogues obtained from these experiments form the basis of our decisions regarding:

- Discourse context
- Syntax extensions
- Vocabulary extensions
- Data base extensions
- Lexical selection of phonetic base forms
- Prosodics

Several protocols were gathered at the Naval Post-Graduate School in Monterey, California, during July, 1974. These were followed by a further protocol experiment in the SDC SUR laboratory. Design of a new set of experiments was worked out with technical and military personnel at the Naval Electronics Laboratory Center (NELC) in San Diego, California. The experiments were conducted with military personnel at NELC in May, 1975. Orthographic transcriptions

of the dialogues will be used to identify necessary syntax, vocabulary, and data base extensions to the system, as well as providing useful information about discourse context. The transcriptions will also serve as prompting material for subjects who will participate in an experiment to be conducted in the SDC laboratory. The results from the latter experiment will be transcribed orthographically at SDC and phonetically at SCRL. These phonetic transcriptions will guide the selection of lexical base forms and their accompanying phonology. Also, the phonetic transcriptions, along with acoustic analyses of the utterances in the dialogues, will be useful in the analysis of prosodic phenomena.

Concordances were found very useful in the analysis of our current protocols. SDC currently has the following capabilities for concordance generation.

- KWIC index for orthographic text (Figure 1-7).
- KWIC index for phonemically transcribed text (Figure 1-8).
- KWIC index for individual phonemes (Figure 1-9).
- A concordance in which keywords are displayed together with the entire sentence in which they appear (Figure 1-10).

All versions provide basic statistics of the text processed, e.g., number of sentences in the text, total number of tokens (words or phonemes, as the case may be), number of types, type/token ratio, frequencies, percentage frequencies, etc.

The orthographic KWIC index (Figure 1-7) for our current protocol files has highlighted frequently recurring sentence types and other grammatical constructions. For example, out of a total of 220 sentences (3 protocols, 3 speakers) 62--i.e., almost a third--begin with "How many <NP>..." and another third begin with "What is <NP>..." and "What's <NP>..." By taking data such as these into account, the parser can focus on the more likely paths first.

The KWIC index routine for phonemically transcribed text is designed to group together all phonetic variants of the same word under its orthographic representation. For example, under "of" (see Figure 1-7) we found the following variants: AX:O, AX:OF, AX:OV, and AX:1V; under "the" we get: DHAX:O, DAH1H:O, DHIY:1, DHIY:2. Such a concordance provides a check on the phonological rules component, whose function is to generate the variants likely to be encountered in a speech situation. It also allows us to select the most commonly used phonetic spelling to be used in the mapper for a first try.

The KWIC index for individual phonemes (in ARPABET transcription) has proved of interest to all those concerned with the acoustic-phonetic processing component of the system. For example, the distribution of vowel contexts for initial plosives, the relative importance of final plosives, and the

NUMBER	OF	REACTORS?	003014
WHICH CLASS	OF	SUBMARINE HAS THE GREATEST NUMBER OF MISSILE LAUNCHERS?	003038
GIVE ME THAT LIST	OF	SUBMARINES AGAIN.	003044
WHAT IS THE SUBMERGED SPEED	OF	THE ALBACORE?	003048
WHAT IS THE SURFACE DISPLACEMENT	OF	THE ETHAN ALLEN?	003005
WHAT IS THE THE LENGTH	OF	THE ETHAN ALLEN?	003011
THE SUBMERGED SPEED	OF	THE ETHAN ALLEN?	003049
THE SUBMERGED SPEED	OF	THE ETHAN ALLEN?	003059
	OF	THE GEORGE WASHINGTON?	003006
AND THE LENGTH	OF	THE GEORGE WASHINGTON?	003012
THE SUBMERGED SPEED	OF	THE GEORGE WASHINGTON?	003060
OF MISSILE LAUNCHERS AND QUANTITY	OF	THE LAFAYETTE.	003002
WHAT IS THE LENGTH	OF	THE LAFAYETTE?	003004
WHAT IS THE SUBMERGED SPEED	OF	THE LAFAYETTE?	003047
WHAT IS THE SUBMERGED SPEED	OF	THE LAFAYETTE?	003058
	OF	THOSE, WHAT IS THE MAXIMUM AND MINIMUM COMPLEMENT?	003068

Figure 1-7. KWIC Index for Orthographic Text

```
OP
      WHAH:2DX IH:0Z DHAX:0 SAX:0BMER:2JH SPIY:2D AX:0 DHIH:0 AE:2LBAX:0KOW:1R                    003048
                     DHAX:0 SAX:0BMER:2JH SPIY:2D AX:0 DHIH:0 IY:2THEN:0 AE:2L(IX/IH):0N         003059
                     DHAX:0 SAX:0BMER:2JH SPIY:2D AX:0 DHAX:0 JOW:2RJH W(AO/AA):2SHIH:0NT(IX/IH):1N 003060
:0 SAX:0BMER:2JH SPIY:2D AE:1N DHAX:0 NAH:2MBUR:0 AX:0F TOW:1RPIY:2DOW:1 TUW:2BZ FUR:0 DHAX:0 JOW:2R 003057
           LIH:2ST AO:2L KLAE:2S(IX/IH):0Z AX:1V AX:0V NOW:2KL(IY/IH):0UR:0 BEL:0IH:2STIH:1K MIH:2S 003001
 DIH:1SPLEY:2SH(IX/IH):0NT LEH:2NXKTH NAH:2MBUR:0 AX:0V TOW:1RPIY:2DOW:1 TUW:2BZ NAH:2MBUR:0 AX:0V R 003002
MBUR:0 AX:0V TCW:1RPIY:2DOW:1 TUW:2BZ NAH:2MBUR:0 AX:0V RIY:1AE:2KTUR:0Z NAH:2MBUR:0 AX:0V MIH:2SEL: 003002
BZ NAH:2MBUR:0 AX:0V RIY:1AE:2KTUR:0Z NAH:2MBUR:0 AX:0V MIH:2SEL:0 L(AO/AA):2NCHUR:0Z  EN:0 KW(AO/AA 003002
:0 L(AO/AA):2NCHUR:0Z  EN:0 KW(AO/AA):2NAX:0TIY:1 AX:0V DHAX:0 L(AO/AA):1FEH:0YEH:2T              003002
          WAH:2DX  (IX/IH):0Z DHIY:2 LEH:2NXKTH AX:0V DHAX:0 LAA:1F(IH/EH):0YEH:2T                003004
DHAX:0 SER:2F(IH/IX):1S DIH:1SPLEY:2SH(IX/IH):0NT AX:0V DHIH:0 IY:2THEN:0 AE:2L(IX/IH):0N        003005
                                                 AX:0V DHAX:0 JHOW:2RJH WAA:2SHIH:0NT(IX/IH):1N  003006
        WAH:20X IH:0Z DHAX:1Q DHAX:0 LEH:2NXKTH AX:0V DHIH:0 IY:2THEN:0 AE:2L(IX/IH):0N          003011
                      AE:2N DHAX:0 LEH:2NXKTH AX:0V DHAX:0 JHOW:2RJH WAA:2SHIH:0NT(IX/IH):1N     003012
                                NAH:2MBUR:0 AX:0V RIY:1AE:2KTUR:0Z                               003014
                                NAH:2MBUR:0 AX:0V MIH:2SEL:0 L(AO/AA):2NCHUR:0Z                  003015
1N HHAE:1Z DHIH:0 GREY:2DX(IX/IH):0ST NAH:2MBUR:0 AX:0V MIH:2SEL:0 L(AO/AA):2NCHUR:0Z            003037
                           WHIH:2CH KLAE:2S AX:0V SAH:2BMUR:0IY:1N HHAE:1Z DHAX:0 GREY:2TIH:1S 003038
R:0IY:1N HHAE:1Z DHAX:0 GREY:2TIH:1ST NAH:2MBUR:0 AX:0V MIH:2SEL:0 L(AO/AA):2NCHUR:0Z            003038
N HHAE:1Z  DHAX:0 GREY:2DX(IX/IH):0ST NAH:2MBUR:0 AX:0V MIH:2SEL:0 L(AO/AA):2NCHUR:0Z            003040
          GIH:2V MIY:1 DHAE:1T LIH:2ST AX:0V SAH:2BMUR:0IY:1NZ AX:0GEH:2N                         003044
1N HHAE:2Z DHAX:0 GREY:2DX(IX,IH):0ST NAH:2MBUR:0 AX:0V TOW:1RPIY:2DOW:1 TUW:2VZ                 003046
      WHAH:2DX IH:0Z DHIY:1 SAX:0BMER:2JH SPIY:2D AX:0V DHAX:0 L(AO/AA):1FEH:0YEH:2T             003047
                     DHAX:0 SAX:0BMER:2JH SPIY:2D AX:0V DHIH:0 IY:2THEN:0 AE:2L(IX/IH):0N         003049
        AH:1 GIH:2V MIY:2 DHIY:2 AH:1 NAH:2MBUR:0 AX:0V MIH:2SEL:0 L(AO/AA):2NCHUR:0Z DHAX:0 SAX:0BM 003057
      WHAH:2DX IH:0Z DHAX:0 SAX:0BMER:2JH SPIY:2D AX:0V DHAX:0 L(AO/AA):1FEH:0YEH:2T               003058
                                                 AX:0V DHOW:2Z WAH:2T IH:1Z DHAX:0 HAE:2RSMAX:1M AE 003068
SAH:2BMUR:0IY:1N HHAE:2Z AX:0 KAA:2MPLAX:0MIX:1NT AX:0V AX:0 HHAH:2BDXUR:0D EN:0 FAY:2V             003069
                LIH:2ST AO:2L KLAE:2S(IX/IH):0Z AX:1V AX:0V NOW:2KL(IY/IH):0UR:0 BEL:0IH:2STIH:1K 003001
```

Figure 1-8. KWIC Index for Phonemically Transcribed Text

```
                                      # DH AX:0 # IY:2 IH EN:0 # AE:2 L IX:0 N #                          3   43
 M ER:2 JH # S P IY:2 D # AX:0 V # DH IH:0 # IY:2 IH EN:0 # AE:2 L IX:0 N #                          3   49
2 D OW:1 # T UW:2 R Z # D AX:0 Z # DH AX:0 # IY:2 IH EN:0 # AE:2 L IX:0 N # HH AE:2 V #              3   54
 B M ER:2 JH # S P IY:2 D # AX:0 # DH IH:0 # IY:2 IH EN:0 # AE:2 L IX:0 N #                          3   59
               # AE:0 N # HH AW:2 # M EH:2 N IY:0 # IY:2 IH EN:0 # AE:2 L IX:0 N Z # AA:1 R # DH UP:1 # 3   16
1 TH # D AX:0 # F IY:2 M # L EH:2 S # DH IX:1 N # IH ER:2 DX IY:0 # F IY:2 T #                       3   62
 AE:2 V # L EH:2 NX K TH # L EH:2 S # DH IX:0 N # IH ER:2 DX IY:0 # F IY:2 T #                       3   63
1 # W IH:1 TH # P IY:2 M # L EH:2 S # DH IX:0 N # IH ER:2 DX IY:0 # F IY:2 T #                       3   64
Z # W IH:1 TH # B IY:2 M # L EH:2 S # DH IX:0 N # IH ER:2 DX IY:0 # F IY:2 T #                       3   65
T UW:2 B Z # D AX:0 Z # DH AX:0 # G AH:2 P IY:0 # IH R IY:2 # HH AE:2 V #                            3   53
```

Figure 1-9. KWIC Index for Individual Phonemes

```
HUNDRED
                3  69   WHICH WHICH SUBMARINE HAS A COMPLEMENT OF A HUNDRED AND FIVE?
               FREQUENCY=      1

I
                1  64    AM I SUPPOSED TO TELL YOU WHAT I CHOSE?
                1  64    AM I SUPPOSED TO TELL YOU WHAT I CHOSE?
                1  66   I CHOSE GEORGE WASHINGTON CLASS SUBMARINE.
               FREQUENCY=      3

I'M
                1  81    I'M SORRY, LESS THAN 30 FEET AND HAS A.
               FREQUENCY=      1

II
                2  43   AND, HOW MANY TORPEDO TUBES AND MISSILE LAUNCHERS FOR THE HOTEL II?
                2  61   WHAT'S THE SPEED OF THE GUPPY III AND GUPPY II, SUBMERGED?
                2  62   WHAT'S THE LENGTH OF THE GUPPY III AND THE GUPPY II?
               FREQUENCY=      3

IIA
                2  58   GUPPY IIA?
               FREQUENCY=      1

IIA'S
                2  59   HOW MANY GUPPY IIA'S DO WE HAVE?
               FREQUENCY=      1

III
                2  57   GUPPY III?
                2  61   WHAT'S THE SPEED OF THE GUPPY III AND GUPPY II, SUBMERGED?
                2  62   WHAT'S THE LENGTH OF THE GUPPY III AND THE GUPPY II?
               FREQUENCY=      3

III'S
                2  60   HOW MANY GUPPY III'S?
               FREQUENCY=      1
```

Figure 1-10. Concordance of Keywords with Sentences

voicing context of /DH/ were items of immediate interest to the researcher working on fricative and plosive analysis. Furthermore, the table ranking phonemes by frequency of occurrence (see Table 1-3) points toward the most pressing areas of research. In the body of protocol sentences analyzed, the problem phoneme N appears at the top of the list, closely followed by /ə/, /s/, /i/, /ɪ/, /z/, /d/, /ð/, /e/. A survey of related work revealed that the phoneme frequency distribution in our protocol sentences compares favorably with the distributions in Denes [4] and Shoup [21]. We therefore feel justified in using the output of this frequency study to guide our acoustic-phonetic research. Particularly, it is important to note that the phones /n/, /ɪ/, /l/, and /r/ have a high frequency of occurrence. They exert a great influence on the vowels in their immediate vicinity. Therefore, research on the coarticulation effects between these phones and neighboring vowels is one of our primary tasks for this year.

1.2.6 Prosodics

A research plan for a joint SDC-SRI study of prosodic features and their use in a speech understanding system has been developed. In this plan, SDC will perform acoustic processing on the dialogues obtained from the protocol experiments. This processing will result in A-matrices that contain parameters such as fundamental frequency, RMS energy values, and formant information. The results will then be analyzed by SDC and SRI. The first experiment deals with word durations and has already begun. A-matrices were prepared for each utterance of a protocol experiment recorded in September, 1974. There were a total of 69 utterances in the protocol. Word and pause durations were determined on the basis of the A-matrix parameters described above. This information was input to a program that created a new file in which each word and pause occurring in the protocol appeared with its duration. For example, utterance #44 in the protocol, which in the original file reads:

"Give me that list of submarines again"

now reads:

"give17 me08 that18 list28 of07 submarines54 again45"

The numbers following each word refer to the number of 10-ms. segments the word spans. For example, the word "give" in this sentence is 17 segments, or 170 ms., long. The new file thus obtained was put through the KWIC indexing routine that groups all similar words together and displays them in their context (see Figure 1-11).

Table 1-4 shows the mean and range of a number of terms that occurred more than 10 times in the corpus. It may be seen that durations for the same term often vary as much as twice their shortest duration. The durations of short function words show much larger variations than those of context words or compound terms.

Table 1-3. Phoneme Frequency Count

Freq.	%Freq.	Phoneme (Computer Representation)	Phoneme (IPA Representation)
1	0.04	EM	syl m,m
9	0.40	AA	ɑ
9	0.40	Q	ʔ
10	0.45	WH	ʍ
12	0.54	NX	ɳ
13	0.58	G	g
14	0.63	UH	ʊ
15	0.67	EN	syl n,n
16	0.72	SH	s or ʃ
17	0.76	AY	aɪ or ay
17	0.76	CH	ʧ
17	0.76	ER	ɝ
22	0.99	F	f
22	0.99	W	w
23	1.03	EY	eɪ or ey
24	1.08	TH	θ
26	1.17	AW	aʊ or aw
26	1.17	UW	u
27	1.21	AO	ɔ
27	1.21	EL	syl l,l
28	1.26	DX	flapped t,ɾ
29	1.30	Y	y
31	1.39	JH	ʤ
33	1.48	K	k
33	1.48	P	p
36	1.62	OW	o
49	2.21	R	r
50	2.25	IX	ɨ
52	2.34	EH	ɛ
54	2.43	HH	h
55	2.48	V	v
56	2.52	AH	ʌ
61	2.75	UR	ɚ
66	2.97	AE	æ
66	2.97	L	l
68	3.06	B	b
76	3.43	DH	ð
78	3.52	D	d
81	3.60	Z	z
90	4.06	IH	ɪ
108	4.87	T	t
114	5.14	M	m
122	5.50	IY	i
134	6.04	S	s
145	6.54	AX	ə
153	6.90	N	n

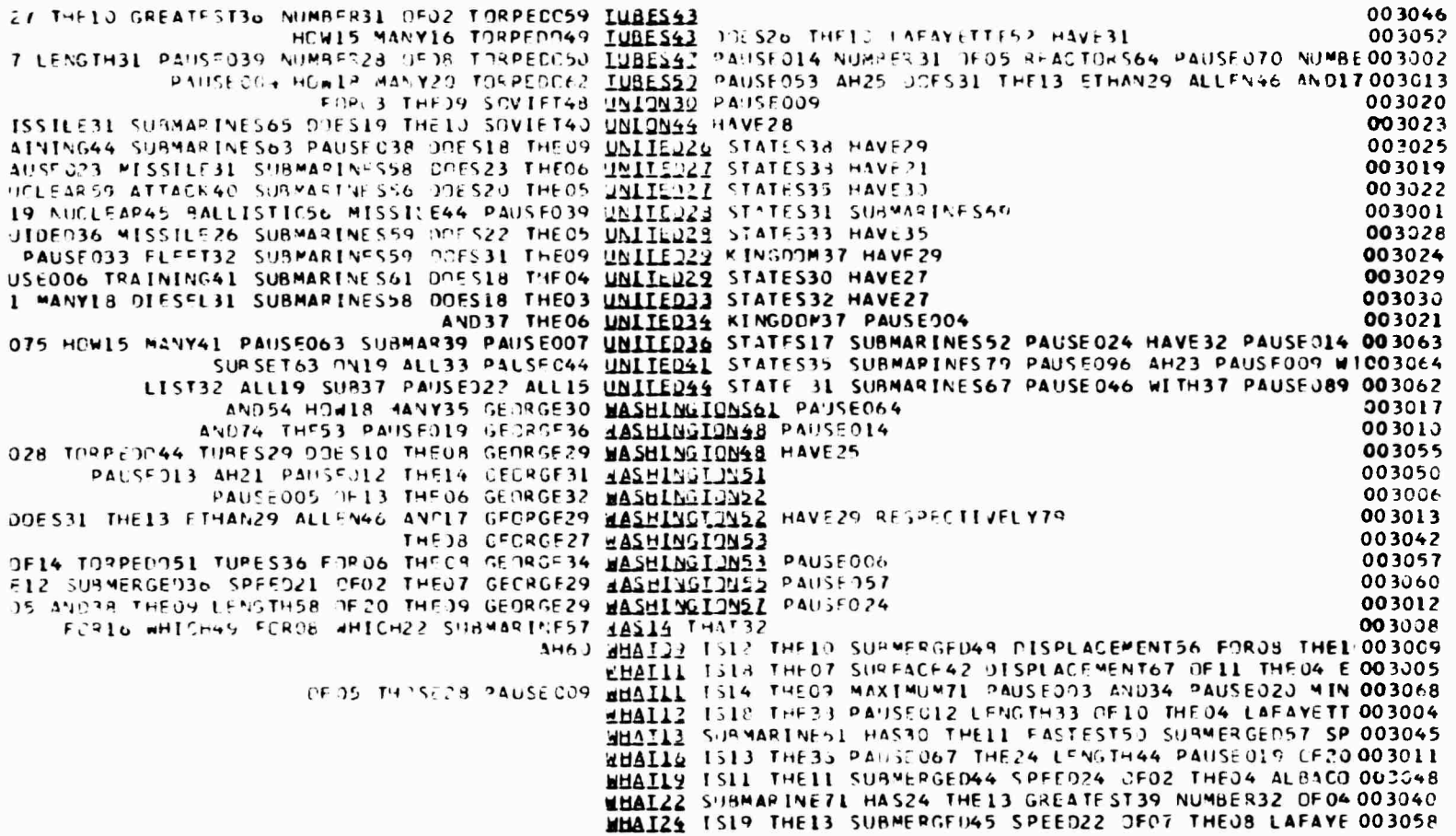


Figure 1-11. Word Durations

Table 1-4. Sample Frequency and Duration Data

TERM	FREQ	MINIMUM DURATION	MAXIMUM DURATION	MEAN	RANGE= MAX.DUR/ MIN.DUR
The	64	3	53	11.67	18 x Min. dur
Of	29	2	31	8.93	15.5
How many	25	24	57	45.12	2.3
Submarines	21	40	91	60.57	2.3
Have	21	15	38	26.9	2.5
Missile	14	26	44	33.64	1.6
Number	11	26	51	33.29	1.96
Submerged speed	8	54	102	74.13	1.89

It is anticipated that duration data of this kind will eventually be used by several components of the system, in particular by the MAPPER.2, where it could become the basis for one of its subsetting functions. It is conceivable that information on word duration could also become part of the user model, along with the speaker-dependent vowel tables and other such data.

Recently, work has begun on a comprehensive prosodic analyzer. The analysis is two-staged. The first stage breaks the utterance into prosodic phrases or breath groups. The second stage then breaks each phrase into syllables.

The phrase analysis begins by breaking each utterance at long pauses. The assumption is made that pauses are most likely to occur at phrase boundaries. The inter-pause blocks are further dissected into the rise-fall contours typical of English phrases. This is done by applying the "hull" function technique advocated by P. Mermelstein (in a slightly different context) to the pitch contour [11,12].

The primary difficulty facing this analysis is the effect of phonetics on nominally prosodic parameters. Flaps and intervocalic voiced plosives can cause dips in the pitch contour unrelated to the phrase structure. Also, the vowel IY, especially following a stop, can cause a rise in pitch that is purely phonetic in nature. The phrase analyzer therefore identifies these phonetic situations by reference to acoustic-phonetic data recorded in the A-matrix and corrects for them.

When completed, the prosodic analyzer will categorize each phrase by contour shape and pitch level. This will permit the system to be aware of such effects as the rising contour characteristic of certain questions, contrastive stress applied to a phrase, and the special pattern that often accompanies the uttering of a list of items. In addition, the higher levels of the system can take advantage of knowing the number of phrases in an utterance to direct attention to utterances of appropriate syntactic complexity.

The second stage of the analyzer, which will break phases into syllables, will be based on algorithms developed by P. Mermelstein at Haskins Laboratories and by W. Lea at Univac. Syllables will be segmented and marked for stress. These data will enable MAPPER.2 to plan its mapping of a word more effectively, and will provide a foundation for bottom driving.

1.3 PLANS

The Milestone System, a prototype of the five-year speech understanding system, will be implemented by September, 1975. The Milestone System will have a vocabulary of 600 words. Acoustic-phonetic and parametric processing will be done by PDP-11/40 and SPS-41 computers. All higher-level linguistic processing and lexical mapping will be done on an IBM-370/145 computer. Continued research in acoustic-phonetics and parameterization will lead to more accurate A-matrices. Further work in lexical mapping, particularly in methods of hypothesizing words purely on the basis of acoustic cues, will enable the system to partially break away from a purely predictive strategy based on linguistic cues. Analysis of protocols will lead to a more comprehensive syntax for the data management task, in addition to providing useful information for building discourse structures of dialogues.

1.4 STAFF

Dr. H. Barry Ritea, Project Leader
James A. Balter
Jeffrey A. Barnett
William A. Brackenridge
Richard A. Gillmann
Iris Kameny
Peter Ladefoged (Consultant)
Lee M. Molho
Douglas L. Pintar
Georgette Silva
Rollin V. Weeks

1.5 REFERENCES

1. Barnett, J. A., "A Phonological Rule Compiler," IEEE Symposium on Speech Recognition: Contributed Papers, pp. 188-192. New York: IEEE, April 1974.

2. ------. A Phonological Rules System. Report No. TM-5478/000/00. Santa Monica: System Development Corporation, January 1975.

3. ------ and D. L. Pintar. CRISP: A Programming Language and System. Report No. TM-5455/000/00. Santa Monica: System Development Corporation, December 1974.

4. Denes, P. B. "On the Statistics of System English," JASA 35(6):892-904, 1963.

5. Fujimura, O. "Syllable as a Unit of Speech Recognition," Proceedings of the IEEE Symposium on Speech Recognition, pp. 148-153. New York: IEEE, April 1974.

6. Gillmann, R. "A Fast Frequency Domain Pitch Algorithm," submitted for publication in JASA.

7. Kameny, I. "Comparison of the Formant Spaces of Retroflexed and Non-retroflexed Vowels," IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-23(1), pp. 38-49. New York: IEEE, 1975.

8. ------, W. A. Brackenridge, and R. Gillmann. "Automatic Formant Tracking," JASA 56 (Supplement):S28 (Abstract), 1974.

9. Kameny, I. and Weels, R. "An Experiment in Automatic Isolation and Identification of Vowels in Continuous Speech," JASA 55:411 (Abstract), 1974.

10. Markel, J. D.; A. H. Gray, Jr.; and H. Wakita. Linear Prediction of Speech--Theory and Practice. SCRL Monograph No. 10. Santa Barbara: Speech Communications Research Laboratory, Inc., September 1973.

11. Mermelstein, P., and G. M. Kuhn. "Segmentation of Speech into Syllabic Units," JASA 55 (Supplement):S22 (Abstract), 1974.

12. Mermelstein, P. "A Phonetic-Context Controlled Strategy for Segmentation and Phonetic Labeling of Speech," Proceedings of the IEEE Symposium on Speech Recognition, pp. 144-147. New York: IEEE, April 1974.

13. Molho, L. M. "Automatic Recognition of Fricatives and Plosives in Continuous Speech Using a Linear Prediction Method," JASA 55:411 (Abstract), 1974.

14. ------. HSI-11B--An ARPANET Interface for PDP-11 Computers. Report No. TM-5434/000/00. Santa Monica: System Development Corporation, October 1974.

15. Newell, A.; J. Barnett; J. Forgie; C. Green; D. Klatt; J. C. R. Licklider; J. Munson; R. Reddy; and W. Woods. Speech-Understanding Systems: Final Report of a Study Group. Pittsburgh: Carnegie-Mellon University, Computer Science Department, May 1971.

16. Öhman, S. E. G. "Coarticulation in VCV Utterances," JASA 39(1):151-168, 1966.

17. Oppenheim, A. V., and J. M. Tribolet. Pole-Zero Modeling Using Cepstral Prediction. Report No. 111. Cambridge: MIT Research Laboratory of Electronics, 1974.

18. Paxton, W. H. "A Best-First Parser," IEEE Symposium on Speech Recognition: Contributed Papers, pp. 218-225. New York: IEEE, April 1974.

19. Ritea, H. B. "A Voice-Controlled Data Management System," IEEE Symposium on Speech Recognition: Contributed Papers, pp. 28-31. New York: IEEE, April 1974.

20. ------. "Speech Input to a Data Management System," Proceedings of the Speech Communications Seminar (SCS-74), Stockholm, pp. 291-298. New York: John Wiley and Sons, 1974.

21. Shoup, J. "Phoneme Selection for Studies in Automatic Speech Recognition," JASA 34(4):397-403, 1962.

22 Walker, D. E. "The SRI Speech Understanding System," IEEE Symposium on Speech Recognition: Contributed Papers, pp. 32-37. New York: IEEE, April 1974.

23. Weeks, R. "Predictive Syllable Mapping in a Continuous Speech Understanding System," Proceedings of the IEEE Symposium on Speech Recognition, pp. 154-158. New York: IEEE, April 1974.

24. Weinstein, C. J.; S. S. McCandless; L. F. Mondshein; and V. W. Zue. "A System for Acoustic-Phonetic Analysis of Continuous Speech," IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-23(1), pp. 54-67. New York: IEEE, 1975.

2. LEXICAL DATA ARCHIVE

2.1 INTRODUCTION

The Lexical Data Archive (LDA) project is addressing itself to the task of providing the ARPA Speech Understanding Research (SUR) projects with semantic and syntactic data for the words in their lexicons. The LDA project provides the following services for each SUR project: it monitors a variety of lexical data sources, selects the data having potential payoff for speech understanding, formats those data for archival purposes, and provides for their dissemination to the appropriate SUR projects. The data in the archive are centered on the 3,000 or so words appearing in the lexicons currently being used by the SUR projects at Bolt Beranek and Newman Inc., Carnegie-Mellon University, and System Development Corporation.

2.2 PROGRESS AND PRESENT STATUS

During the first contract year, substantial progress was made toward accomplishing our primary task--the basic design of the archive, which is called the Semantically Oriented Lexical Archive (SOLAR).* The methodology of construction decided upon has been implemented. Files with a significant amount of high-quality data are now accessible via the ARPA Network. Data have been distributed upon request to more than 65 researchers across the nation and abroad. Implementation was begun for the first eight of the following ten files.

(1) A word index, which allows a user to easily determine the words for which data are being collected and the types of data currently available for a given word.

(2) A bibliographic reference file, intended primarily as a resource for accessing the literature.

(3) A file of semantic analyses, which contains formal treatments of the semantic properties of individual words as found in the literature.

(4) A file summarizing the theoretical backgrounds of the technical documents from which the semantic analyses have been extracted.

(5) A file explaining and commenting on the semantic components used in the semantic analyses.

---

*This included the decision as to the types of lexical data to be collected, the determination of the data collection procedures, the specification of programs needed to extract data from machine-readable transcripts, and the design of the logical structure of the files to be built.

(6) A file of integrative summaries of conceptual analyses given in the literatures of philosophy and artificial intelligence for notions coinciding with or underlying the semantic components.

(7) A file of collocational information extracted from definitions in Webster's Seventh New Collegiate Dictionary (W7).

(8) A keyword-in-context (KWIC) file containing every context of each SUR word as found in the W7 definitions, in the Brown Corpus, and in selected speech dialogues.

(9) For each SUR lexicon, a subfile of definitional links between words within that lexicon. These subfiles are being constructed so that a SUR researcher can observe the semantic interrelations in his lexicon.

(10) A file of semantic fields, which are being designed for each SUR word by tying to it words found in certain definitional, synonymitive, and antonymitive relationships in W7, Webster's New Dictionary of Synonyms (WNDS), and/or Roget's International Thesaurus (Roget).

For a more detailed discussion of the contents of each of these files, see Diller et al. [1-8,11].

Since October, 1974, we have concentrated on the following tasks:

- collection of analyses from the literatures of linguistics and analytic philosophy,
- development of computer programs for discovering and displaying links among words in particular lexicons,
- development of a file-management facility, and
- distribution of data to the SUR projects.

The following sections provide more detail on our progress in working on these four tasks.

2.2.1 Collecting Data

Data collection has continued unabated for the hand-built files (files 1-6, above). Approximately 150 semantic analyses have been extracted, and these, together with about 250 prepared previously, have been converted into machine-readable data sets and entered into the on-line data base. The file of semantic analyses now comprises about 375 analyses pertaining to about 275

different words. SOLAR contains relatively thorough coverage of two lexical areas highly relevant to the SUR tasks: prepositions and words indicating spatial and temporal dimensions.

Explanations of about 100 semantic components used in the new semantic analyses have been written and computerized, bringing the total to approximately 215.

During the current year, in response to a suggestion by SRI, we added a file indicating the theoretical orientations of the authors from whose writings the semantic analyses are extracted. The file now contains approximately 20 entries. It serves two primary purposes. First, it reduces the repetition of theoretical orientation formerly inserted in the preliminary qualifications of each semantic analysis and allows a much more readable and complete introduction to an author's perspective. Second, it allows explanations of an author's notational conventions to be included with his theoretical orientation.

The file containing integrative summaries of conceptual analyses has received considerable attention in recent months. The principal innovation in our technique of preparing integrative summaries has been the decision to include a formalization of the essential points presented in each summary. The formalizations, which are stated in a language based on a higher-order predicate calculus with set theory, facilitate the exploitation of the file for computational purposes. We have found that the preparation of formalizations tends to bring to light many unsuspected interconnections among the assertions in the integrative summaries, in some cases revealing inconsistencies that we have been able to eliminate. In addition, the formalizations, as extremely condensed yet precise versions of the summaries, have the effect of turning the other parts of the summaries into background material. Especially in the case of the longer integrative summaries, this is clearly a very desirable result.

Having determined that most of the time we have devoted to preparing integrative summaries has been spent reading the various treatments of a given notion in the recent literature of analytic philosophy, we decided to solicit integrative summaries for particular notions from graduate students in philosophy by organizing a contest. The rationale for the contest is that a student who is already familiar with recent conceptual analyses of a given notion should be able to prepare an integrative summary for it with little effort. We are about to mail to all Ph.D.-granting departments of philosophy an announcement offering a cash prize for the best integrative summary submitted for any of the notions listed in the announcement. If the contest brings in as many high-quality summaries as we hope it will, we will repeat it for additional notions.

The file of bibliographic entries has been augmented by about 2,000 new entries, bringing the total to more than 5,000. About 700 of these pertain to documents in the area of lexical semantics; i.e., documents that contain analyses of particular words or propose universals for lexical representation. More than

2,500 citations relate to documents in experimental phonetics, psychoacoustics, speech analysis and synthesis, psycholinguistics, and phonology. The remainder of the entries pertain to theoretical issues in syntax, semantics, and the philosophy of language or deal with natural-language processing systems and other topics in the field of computational linguistics.

2.2.2 Developing Programs

The file containing collocational features extracted from W7 is complete for the words currently in the SOLAR index. Nearly 1,000 words have one or more features associated with them; i.e., there is some subject label, verbal illustration, parenthetic phrase, or usage note.

The programs needed to produce the KWIC data sets for the Brown Corpus are now complete and checked out. We are temporarily holding up production runs while disk allocations are worked out for the ARPA Network interface. Processing of the W7 contexts is being delayed intentionally until the utility of the Brown contexts is evaluated.

Considerable progress was made in creating the programs and data sets needed to build the file displaying definitional links between words in particular lexicons. The data sets being built comprise the words occurring in the SUR lexicons, the syntactic parts of speech for each, the W7 definitions for each, words standing in an inflectional relationship to the words in the SUR lexicons, and a list of stop words for which no definitional links are followed.

Work on the file of semantic fields has centered mainly on the definition of a data structure and the collection of relevant data sets. The programs themselves have been roughed out, and coding will begin when the definitional expansion programs are completed. Keypunching of the antonym relations found in WNDS began in October, 1974, and is now complete.

2.2.3 Providing ARPA Network Access

Five of the files that are currently computerized are accessible via the ARPA Network. Considerable effort was spent during the first half of this contract year in moving all SOLAR data to the current data management system. The logical structure of all SOLAR files was re-evaluated and revised where necessary, and the programs needed to convert the data sets to the revised format were coded and run. The ARPA Network interface is currently being tested and refined.

LDA is presently attempting to familiarize its potential users with the SOLAR data structures, contents, and accessing protocols. Comprehensive user's guides have been completed for some of the files and are being written for the remainder. Single-page introductory sheets are being distributed to prospective users of the ARPA-Network access mode. Documents describing the

archive have been distributed from SDC [6], have been published in Computers and the Humanities [7], and have been accepted by the American Journal of Computational Linguistics [8]. Trips to the SUR groups to demonstrate SOLAR and the accessing protocols are also planned.

2.2.4 Distributing Data

In October, 1974, listings of the word index, the bibliographic citations, the semantic analyses, the semantic components, the conceptual analyses, the collocational features, and portions of the KWIC file were distributed to the SUR groups and approximately 30 other researchers in semantics.*

Since SOLAR's initiation, requests for information on SOLAR have been received from more than 60 sources. In April, 1975, SOLAR first became accessible via the ARPA Network, and user's guides were distributed.

2.3 PLANS

During the remainder of this contract year, the LDA staff will focus on four tasks. First, we will continue to collect data from the literature. This will involve extending the bibliographic and semantic and conceptual analysis files, more than doubling the theoretical backgrounds file, adding to the semantic component file, and updating the word index. The files containing semantic analyses and semantic components will increase in size as documents already in hand are processed, as new documents are found, and as new SUR words are added. The number of integrative summaries of conceptual analyses will increase as we strive for a more complete coverage of the semantic components already entered in SOLAR and as semantic analyses involving new semantic components are added. The file of collocational features will be automatically updated as new SUR words are added.

Second, we will continue program development. The collocational feature and KWIC files may be restructured, and some programs may be revised as a result of feedback from users regarding the utlity of each file. We will also be coding and running the programs needed to produce the two remaining machine-derived files (i.e., the file linking words definitionally and the file of semantic fields). The semantic field file will in all likelihood be quite rudimentary and still very incomplete by the end of the present contract year. Because of the great amount of logical restructuring, programming, and computer processing required for the development of this file, as well as its practical and theoretical significance, we have singled it out as an area of special focus in the next year.

---

*The data distributed included about 150 semantic analyses together with explanatory notes and 11 integrative summaries of conceptual analyses.

Third, we will continue to add SOLAR files to those now available on the ARPA Network and distribute user's guides explaining on-line accessing procedures. Demonstrations of accessing procedures and file utility will also be given at the SUR sites. Fourth, we will continue to disseminate data from each of the files.

In the coming (1975-76) contract year, we will pursue seven tasks. First, we will continue hand collection of data for the bibliographic reference, semantic analysis, and conceptual analysis files. Even the most advanced of these files will be far from complete by the end of the current contract year.

Second, we will update each of the files on the basis of the new words added to the SUR lexicons.

Third, we will improve the semantic field file. Work will still remain in the following four areas: (1) optimizing the overall design of the file (contents and structure), (2) completing the collection of appropriate data sources, (3) processing these sources, and (4) displaying the relevant data in manageable quantities. The structure and accessing protocols will be revised in accordance with (a) suggestions from users as to how the utility of the file could be enhanced and (b) our review of the semantic networks constructed by the SUR groups (cf. Nash-Webber [10] and Hendrix [9]).

Fourth, we will add to the KWIC indices the data in the SUR protocols. Addition of the SUR protocols will require two types of activities: (1) the construction of standardized machine-readable files, and (2) the collection and modification or creation of programs that will produce useful printouts. Step 1 includes designing a format that allows the tagging of questions, commands, parenthetic comments, narrative input, answers, feedback, etc. Actual file production will involve keypunching, data conversion programs, and on-line editing. Step 2 will be facilitated by the fact that some of the needed programs already exist at SDC and, with only minor modification, can serve as production vehicles. The rather sophisticated KWIC programs developed for the processing of medical titles (Olney, et al. [12]) will provide frequency data for (even very long) recurring phrases as well as for single words, together with data pertaining to their inclusion relationships. KWIC data of these kinds can be provided for individual protocols or collections of protocols, as desired.

Fifth, we will refine the data management output to tailor it more directly to the specific SUR projects requesting data. This includes improving the selection capabilities in data retrieval, increasing control over printout of the data retrieved, and--for specifically requested terms--adding original linguistic research findings to the hand-collected files.

Sixth, we will clean up network interface problems, make any remaining SOLAR files accessible over the network, and assure a smooth and useful file transfer capability for data selectively retrieved from SOLAR.

Seventh, the desirability of moving SOLAR to the Datacomputer will be evaluated on the basis of such considerations as the relative importance of the current lack of report generation capabilities in the DMS being used and the restriction to a single level of repeating group structure.

2.4 STAFF

Dr. Timothy C. Diller, Project Leader

Thomas Bye (part-time)
Enrique Delacruz (part-time)
Frank Heath (part-time)
John Olney (Consultant)
Nathan Ucuzoglu (part-time)

2.5 REFERENCES

1. Bye, T.; T. Diller; and J. Olney. User's Guide to the SOLAR Semantic Analysis File. Report No. TM-5292/001/00. Santa Monica: System Development Corporation, April 1975.

2. Diller, T. User's Guide to the SOLAR Bibliography File. Report No. TM-5292/000/01. Santa Monica: System Development Corporation, December 1974.

3. Diller, T., and T. Bye. User's Guide to the SOLAR Theoretical Backgrounds File. Report No. TM-5292/002/00. Santa Monica: System Development Corporation, April 1975.

4. Diller, T.; T. Bye; and J. Olney. User's Guide to the SOLAR Semantic Component File. Report No. TM-5292/003/00. Santa Monica: System Development Corporation, in preparation.

5. Diller, T., and F. Heath. User's Guide to the SOLAR KWIC File. Report No. TM-5292/008/00. Santa Monica: System Development Corporation, May 1975.

6. Diller, T., and J. Olney. SOLAR: A Semantically Oriented Lexical Archive. Report No. SP-3726. Santa Monica: System Development Corporation, November 1973.

7. ------. "SOLAR (A Semantically Oriented Lexical Archive): Current Status and Plans," Computers and the Humanities 8(5-6):301-312, September-November 1974.

8. ------. "SOLAR: A Comprehensive Source of Semantic Lexical Data," American Journal of Computational Linguistics, forthcoming.

9. Hendrix, G. "Current State of Semantics for the SRI-SDC Speech System." Menlo Park: Stanford Research Institute, unpublished paper.

10. Nash-Webber, B. Semantics and Speech Understanding. BBN Report No. 2896. Cambridge: Bolt Beranek and Newman Inc., 1974.

11. Olney, J.; E. Delacruz; T. Dilier; and N. Ucuzoglu. User's Guide to the SOLAR Conceptual Analysis File. Report No. TM-5292/004/00. Santa Monica: System Development Corporation, June 1975.

12. Olney, J.; R. Weeks; and B. Yearwood. "Vocabulary Control via Automatic Extraction of All Recurring Content-Word Phrases from KWIC Indexes," Proceedings of the ASIS 10:171 ff., 1973.

3. COMMON INFORMATION STRUCTURES

3.1 INTRODUCTION

The need to share data for multiple applications, and the need to move existing data bases to new systems, make general techniques for data base conversion desirable. This need is especially apparent when data are created and manipulated by increasingly complex data management systems. The goal of the Common Information Structures project is to develop techniques for data base conversion that are practical for application to current data management systems and that are designed to be easily used by data base users.

The difficulties in converting a data base from one data management system (DMS) to another arise from the fact that data base structures are system and application dependent. Data bases generated by data management systems (DMS) are organized in the computer in ways designed to achieve certain efficiencies. These efficiency considerations, determined by total cost, response time requirements for data retrieval, and storage trade-offs, depend on the projected application of the data. Accordingly, DMSs utilize data structures designed specifically for their particular application. For example, TDMS (an SDC DMS), which was designed to achieve on-line fast retrieval for unpredictable queries, uses a fully inverted data structure that results in an inflation of the physical data structures by a complex array of tables and pointers. MARK IV (an Informatics DMS), on the other hand, is primarily oriented to generating reports in batch mode. Consequently, data are organized physically using simpler mechanisms (sequential and index sequential files). These examples illustrate that data base conversion involves not only the logical structure of the data as the user views it, but also the storage structures, which reflect specific efficiency requirements, and the physical structures, which reflect computer system hardware.

The conventional approach to converting data bases for new applications is to write a special-purpose conversion program for each data base. This is a very expensive process, since it requires a different program for every source-to-target data base conversion. Furthermore, the programmer needs to account for the three levels of internal structuring mentioned above. This requires detailed knowledge of storage and physical information for the particular data management systems involved.

Another possible approach is to define data description languages for all three levels of structure, then specify in these languages the source and target data bases, as well as conversion statements between them. A detailed discussion of this approach and some of the papers that describe it is given in [1]. Since this approach involves all three levels, it requires complex and detailed data description languages, which are difficult to learn and to use. It also requires that data be converted from the source physical environment to the corresponding target physical environment, which further complicates any possible implementation.

As shown in Figure 3-1, the conversion system has three principal components: (1) a source reformatter, which reformats the output of the source DMS into a standard data form; (2) a translator, which logically restructures the data from the source standard form to a target standard form; and (3) the target reformatter, which reformats the target standard data into an input data stream for the target DMS. The reformatting process does not involve any logical restructuring of data; rather it is a one-to-one mapping of values. Our approach simplifies the data conversion process in that it requires a data translator that operates only on logical data, and two conceptually simple data reformatters. To facilitate the automation of data conversion, we use three languages:

1. A Common Data Description Language (CDDL). This language describes only the logical properties of data bases, and is, therefore, quite simple. One can describe in it how fields are grouped together, the relationships between groups, and field properties.

2. A Common Data Translation Language (CDTL). This language was designed to express logical restructuring functions primarily in terms of field-to-field mappings. This design makes the language easy to use, since field-to-field mappings are simple concepts to specify. This includes repetition and elimination of field values, creation and elimination of group levels, and modification of data values. In addition, one can describe concatenation of several source fields into one target field, subset the records to be converted, and order the records after conversion. A more detailed description of these functions is given in the next section.

3. A Common Data Format Language (CDFL). Statements in this language will be used by the reformatting processors at both the source and target ends. In this language, one can specify the input and output format conventions used by the target and source DMSs, respectively.

Source and target data descriptions are required for every data base conversion, as well as translation statements associating source and target elements. On the other hand, data format descriptions are required only once for every data management system.

The central component of the system is the translator, which accepts statements in CDDL for the source and target data bases, as well as statements in CDTL that describe mappings and associations between source and target data fields. The translator consists of two main components: the analyzer and the converter. The analyzer performs syntax analysis on the CDDL and CDTL statements, and semantic analysis to determine what correspondence is implied between the source and target groups and whether translation requests are legal. The converter uses a conversion table generated by the analyzer to convert source records into target records.

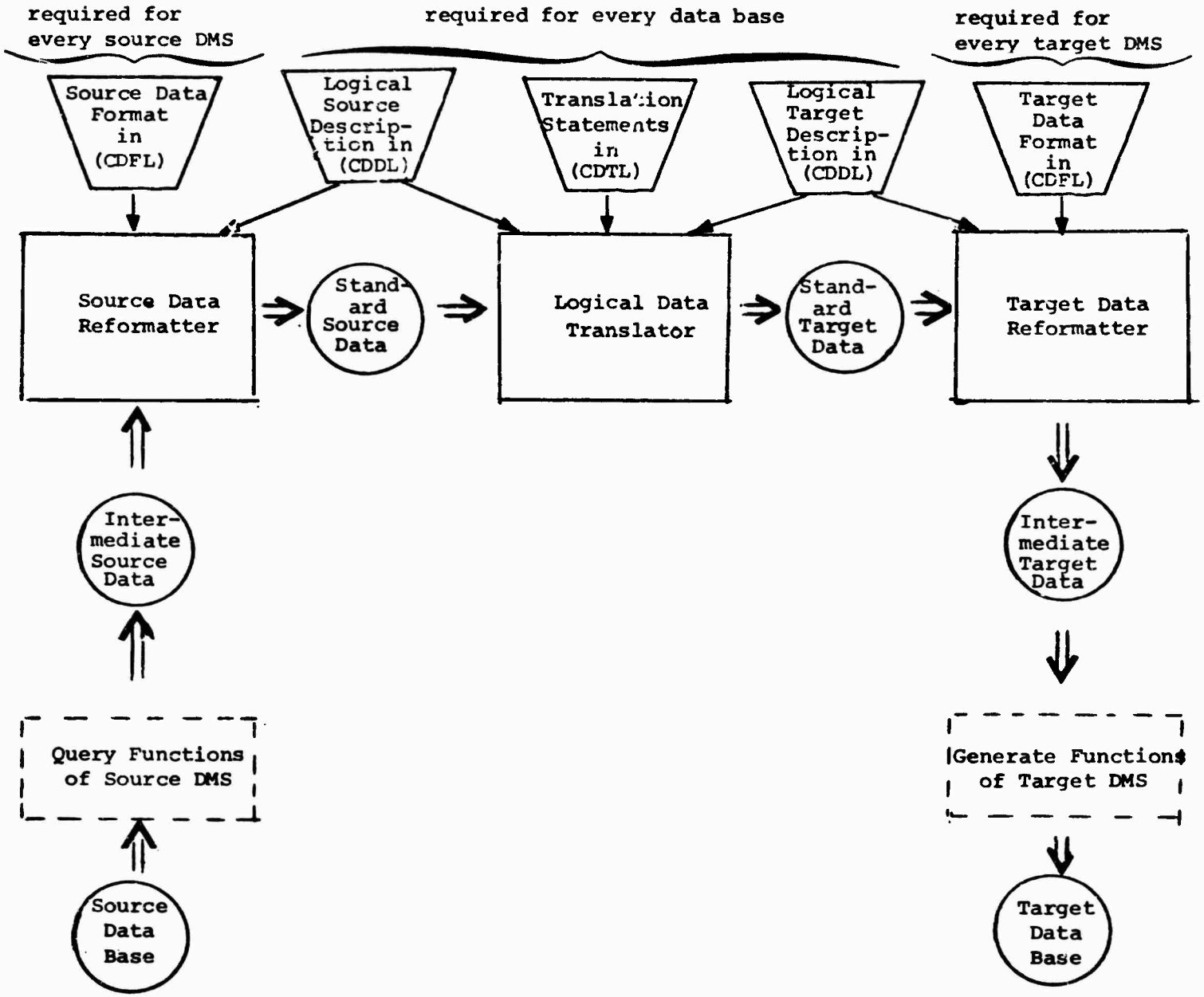


Figure 3-1. The Data Conversion Process

3.2 PROGRESS AND PRESENT STATUS

Major progress was achieved during the first half of this contract year in the development of the translator. The analyzer was designed and partially implemented; the converter was designed, implemented, and debugged. A few sample data bases were successfully converted, testing the different components of the converter.

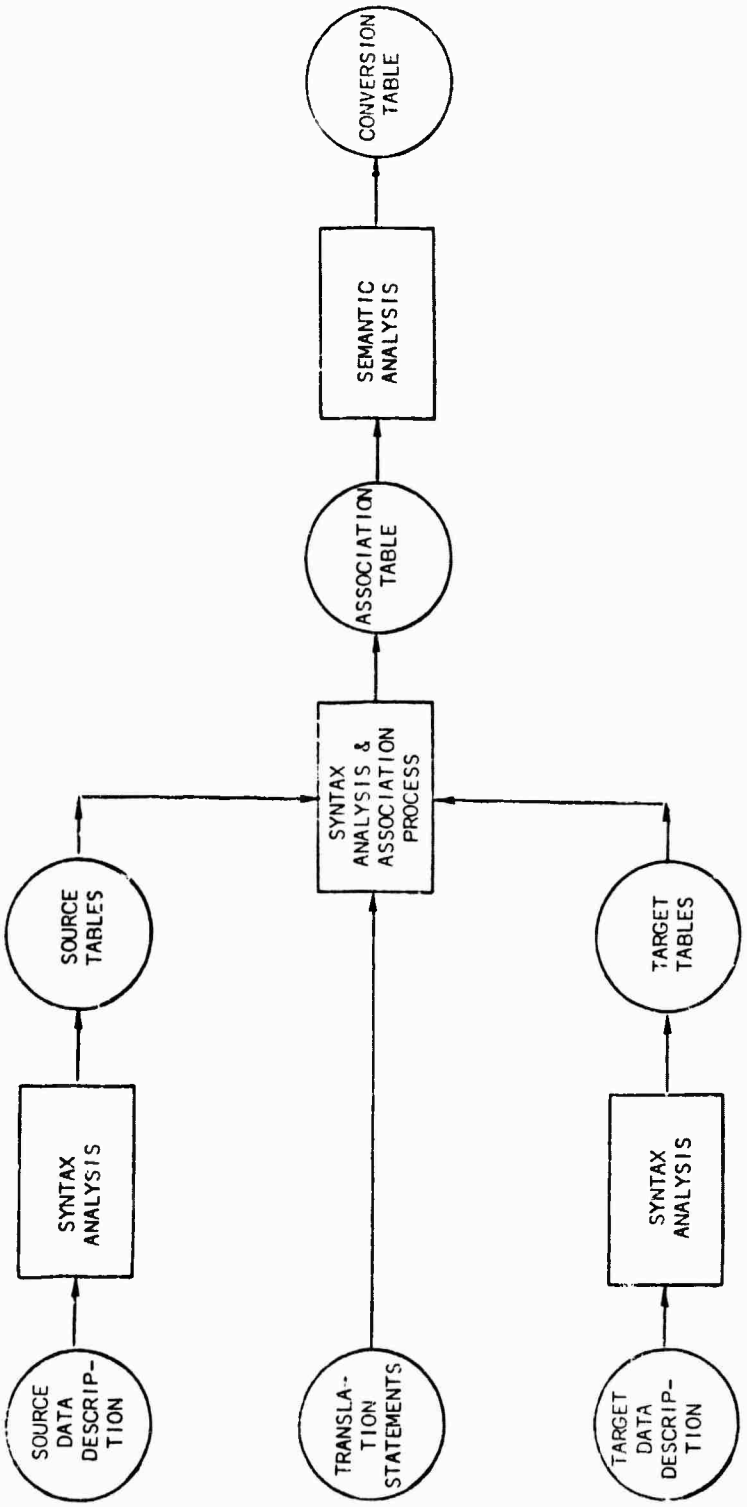
3.2.1 The Analyzer

The analyzer is diagrammed in Figure 3-2. First, syntax analysis is performed on the source and target data description statements (in CDDL). If no errors are found, source and target tables are produced that contain precise information about the data bases. If an error is found, an appropriate error message is issued. Errors discovered at this stage may be more than strictly syntactical; for example, the description of a field may be missing. The next step is the association process, which associates source and target fields according to the translation statements. In this step, the translation statements are also checked for syntax legality. This process produces the association table, which is used by the semantic analyzer.

The conversion functions, which are expressed in CDTL, are designed to achieve ease of use and simplicity of expression. These objectives are achieved by using mainly field-to-field mappings between fields of source and target groups. The types of mappings (or "conversion functions") that we have found useful are summarized and explained in Appendix A.

A central concept in determining the meaning of a conversion function is the correspondence between a source group and a target group. We say that a source group corresponds to a target group if, for every target group instance, there exists a unique instance in the source group. The purpose of the semantic analyzer is to determine, from the collection of conversion functions requested by a user, whether the request is semantically meaningful. To achieve this, the semantic analyzer examines this collection for possible conflicts and, if none are found, determines the correspondences between source and target groups. Conflicts arise if two or more conversion functions cannot coexist according to a predetermined set of semantic rules. For example, if a DIRECT function exists between a source group and a target group, no conversion function of a different type can exist between these two groups. These rules follow from the nature of the conversion functions. A complete list of the semantic rules and their implications is given in Appendix B.

After the semantic analysis is found correct, and the correspondences between source and target groups have been determined, the conversion table is constructed. Every entry in the conversion table contains a coded instruction to the converter to perform one of the functions required (such as DIRECT, REPEAT, GROUP). The entry includes information about the source field from



Figre 3-2. The Analyzer

which a value (or values) is to be extracted, the target field to be created, the conversion function required, and additional operations (such as "string modification" and "subset") if specified. The conversion table is the sole input to the converter.

3.2.2 The Converter

There are three basic conversion types, distinguished by the correspondence between the source and target groups. We label them as the DIRECT, LEVELUP, and BUNDLE types. In the DIRECT type the dominant conversion function is DIRECT and the LEVELUP and BUNDLE functions are not permitted. All other conversion functions are allowed with this type. In the LEVELUP type, the target group created has no direct correspondence to a source group, since it is created by a LEVELUP function. According to the semantic constraints (rules), only a subset of other conversion functions are permitted with a LEVELUP type. Similarly, in the BUNDLE type, there is no direct correspondence to a source group, and only a subset of the functions are permitted.

There is a radical difference in the implementation of these three types, and we are proceeding to implement them separately. The first type to be implemented is the DIRECT type, because it is the most common one and permits the use of most of the conversion functions.

The DIRECT type is diagrammed in Figure 3-3. It is basically table driven by the conversion table (CTAB) and keeps track of the current CTAB entry. As it proceeds, it also keeps pointers to the current instances of both the source and target data for all the levels of the hierarchy involved.

The controller reads the current CTAB entry to determine which module to call. The DIRECT, REPEAT, INSTANCE, OPERATION, and AS-IS modules in turn call the READER module (possibly more than once) to extract the desired value(s) from the appropriate level of the source hierarchy. The CONCATENATE module can call on other modules to extract the values to be concatenated. Then, the value returned to the controller is written into the target record by the WRITER module. The GROUP and END modules are responsible for repositioning the current CTAB entry, and the current pointers to the source and target data, when a new (lower-level) target group is to be formed or the current group is to be "closed." Some of the modules mentioned above can call additional modules to perform lower-level functions such as string modification or subset.

The controller continues to move up and down the CTAB entries until all source instances have been exhausted. Then it gets the next source record and repeats the operation. When all source records have been processed, the conversion process terminates.

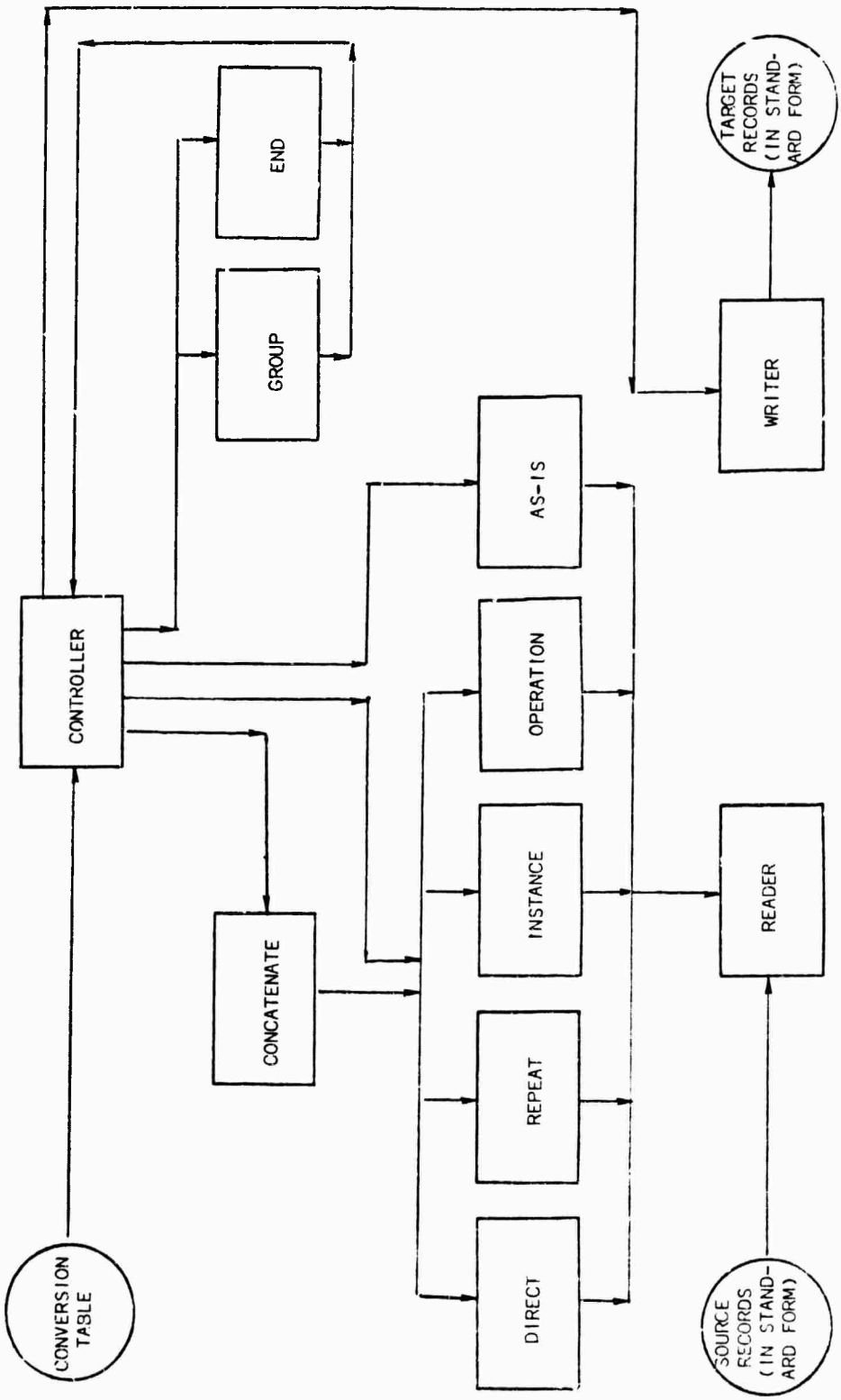


Figure 3-3. The Converter

3.3 PLANS

Our plans for the rest of this contract year fall into three categories.

1. The completion of the translator. This includes completing the design and implementation of the analyzer and debugging it. The analyzer will be designed so that all conversion types will be semantically checked. In addition, the implementation of the direct converter must be completed by adding a few lower-level modules, such as string modification, number type conversion, and others. All components will then be checked thoroughly with several example data bases.

2. A preliminary investigation and design of the reformatting functions. Here we need to study different output forms of DMSs and design a language that reflects them. A decision whether one general-purpose reformatter should be built or several different reformatters (for different output forms) will be made at this time.

3. A simple experiment of data base conversion on the ARPA Network. We will attempt to convert a small data base from the ARPA-DMS to the Datacomputer. Since the reformatters will not exist at this time, the reformatting process will be hand-tailored for these data bases, allowing the exercise of the translator.

3.4 STAFF

Dr. Arie Shoshani, Project Leader
Kenneth M. Brandon

3.5 REFERENCES

1. Shoshani, A. "The Logical Approach to Data Base Conversion," Proceedings 1975 ACM-SIGMOD Workshop on Management of Data: Description, Access and Control, San Jose, May 1975.

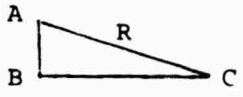
APPENDIX A. TYPES OF CONVERSION FUNCTIONS

The functions explained below form the basis for the Common Data Translation Language (CDTL).

I. DIRECT--allows for a one-to-one mapping between a source field and a target field belonging to groups that correspond. In addition, a modification function can be expressed to perform value modifications, such as truncation, removing excessive blanks, etc. The modification functions are quite powerful, but can be interpreted sequentially by the translator at conversion time.

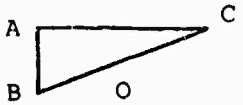
II. REPEAT--Suppose A and B are source groups where B is subordinate to A (by one or more levels). Also, suppose that B corresponds to a target group C. Then the REPEAT function is necessary when a field value in A is to be repeated throughout instances of C. As in the DIRECT case, a modification clause can be used with REPEAT.

The following triangle



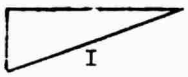
is a symbolic notation for a REPEAT where a horizontal line signifies a correspondence, the vertical line a subordination of groups, and the directional labeled line a mapping of type R (repeat). We will use this notation for other functions as well.

III. OPERATION--Suppose that B is a source group subordinate to a source group A, and A corresponds to a target group C. The OPERATION mapping allows a set of values in instances of B to be combined by some operation (e.g., average, count, maximum) into a single value for a field in C. The following triangle

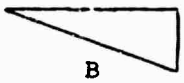


represents an operation. A subset clause permits a selection of only part of the source instances.

IV. INSTANCE--This function is similar in nature to OPERATION except that one can select a particular instance of the source group, such as "the oldest child" or "the n^{th} instance." The selection criteria are expressed with the selection clause, and a value modification is permitted after obtaining the desired value. Symbolically, a similar triangle to the OPERATION case represents INSTANCE:



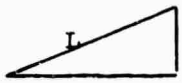
V. BUNDLE--This function effects the creation of a new target group by combining multiple field values in a source instance into group instances of the target group. For example, if a source employee group had several fields for children, then those values could be combined to form a children group in the target. This function is desirable when the source DMS lacks enough hierarchical levels (such as NIPS/FFS). Symbolically, a BUNDLE is represented as the following triangle:



showing that there is a correspondence between the source group and a target group above the one created.

VI. LEVELUP--This is another case of a new target group creation. In this case, the source group contained repetition of field values, and we wish to get rid of the repetition by the creation of a new target group level. For example, if the source group is about "projects" which have a department field repeating for projects in the same department, then it might be useful to create a target

data structure where departments form a group and projects form a subordinate group. The triangle representing LEVELUP is



showing that there is a correspondence between the source group and a target group below the one created.

VII. CONCATENATION--This function allows the user to specify that a target field is to be made up from two or more values extracted from the source. For example, we may wish to concatenate last name of employee obtained with a REPEAT function and first name of child obtained with a DIRECT function.

VIII. AS-IS--This is simply a way to specify that a collection of fields and/or groups are to be moved unchanged to the target data. This is a short way of expressing many DIRECTs which have no modification. In fact, the translator checks for consecutive DIRECTs and combines them into one global operation. The standard structure was designed as a linearized tree with relative instance pointers so as to permit the transfer of "chunks" of data in consecutive fields. We think that the AS-IS function would be used very extensively and we designed the translator and the standard form to perform this function efficiently.

IX. SUBSET--This function provides a tool for presubsetting the source records before translation takes place. For example, only employee records in the R/D department are to be considered. We do not see the need for subsetting within records.

X. ORDERING--A facility for post-translation ordering of records is provided. Ordering within records can also be useful.

Other conversion types were considered (such as splitting a source group into two subgroups), but they were judged to be less useful and, therefore, are not included here.

APPENDIX B. SEMANTIC ANALYSIS

Once the desirable functions were determined, the question of which combination of these functions should be permissible arose. The concept of correspondence, together with the nature of the functions, led to a set of semantic rules that determine correct combinations of mappings as follows:

Rule 1: At least one DIRECT must exist between groups that correspond. Inversely, if a DIRECT exists between a source field and a target field, the source and target groups involved must correspond. This rule is a direct consequence of the definition of correspondence.

Rule 2: For OPERATION, INSTANCE, REPEAT, BUNDLE, and LEVELUP, the "triangles" (see Appendix A) must exist. From Rule 1 it follows that a DIRECT must exist where a correspondence (horizontal line) exists. This rule follows because of the nature of these functions, as one can verify easily. There is a minor exception in the case of LEVELUP and BUNDLE where a virtual correspondence can exist because of the creation of new target groups.

Let us classify functions into three types according to their direction: "straight" (DIRECT), "up" (INSTANCE, OPERATION, LEVELUP), and "down" (REPEAT, BUNDLE), then:

Rule 3: Mappings between the same source group and the same target group must be in the same direction. For example, a REPEAT and a DIRECT cannot coexist, while a REPEAT and BUNDLE can. Of course, many functions of the same type, such as many DIRECTs, can coexist. This rule also follows from the nature of the functions. This rule can be summarized in the following table (where INSTANCE and OPERATION are represented jointly as V--for "value function"--because of their similar nature):

*: acceptable combinations

	D	R	V	B	L
D	*				
R		*			
V			*		
B		*		*	
L			*		*

Rule 4: When considering multiple functions to the same target group from different source groups, some combinations are not permissible as shown in the following table:

*: permissible combinations

	D	R	V	B	L
D					
R	*	*			
V	*	*	*		
B		*			
L		*	*		

Some simply are logically impossible, such as DIRECT and BUNDLE, and some indicate that the source structure was ill-formed, such as OPERATION and BUNDLE.

Let us define the level number L(X) for a group X as the group "distance" of this group from the top of the hierarchy.

Rule 5: Suppose that a source group X corresponds to a target group Y. Let X' be the source group which corresponds to the predecessor of Y. Then

$$L(X') < L(X)$$

This rule is necessary to ensure that no inconsistent cross mappings are specified. The exception here is the case of INVERSION, which must be requested specifically as such.